

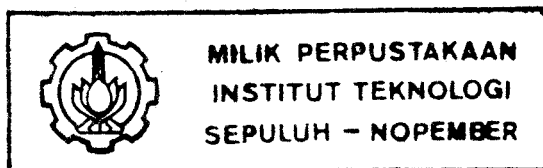
3100096007348

**SIMULATOR CENTER OF GRAVITY AND WEIGHT
MANAGEMENT SYSTEM (CG&WMS)
PADA PROGRAM UJI TERBANG DENGAN
MENGUNAKAN TRANSPUTER T805**

PERPUSTAKAAN ITS	
Tgl. Terima	22 SEP 1994
Terima Dari	H
No. Agenda Prp.	2723



RSE
629.132.36
Suw
S-1
1994



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

OLEH :

EDI SUWARTONO

2882201108

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA**

1994

**SIMULATOR CENTER OF GRAVITY AND WEIGHT
MANAGEMENT SYSTEM (CG&WMS)
PADA PROGRAM UJI TERBANG DENGAN
MENGUNAKAN TRANSPUTER T805**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro**

Pada

Bidang Studi Elektronika

Jurusan Teknik Elektro

Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui / Menyetujui

Dosen Pembimbing I



(Ir. Soetikno)

NIP. 130445231

Dosen Pembimbing II



(Ir. Hendra Kusuma)

NIP. 131846104

SURABAYA

Agustus, 1994

ABSTRAK

Center of Gravity (CG) atau titik berat merupakan salah satu parameter yang perlu diketahui selama pesawat terbang beroperasi. Pada saat beroperasi berat pesawat terbang akan berkurang akibat pemakaian bahan bakar. Dengan berkurangnya berat pesawat terbang maka posisi titik beratnya akan berubah pula.

Perubahan posisi titik berat akan mempengaruhi kestabilan pesawat terbang, karenanya diperlukan pengontrol untuk mengantisipasi perubahan titik beratnya. Dalam tugas akhir ini akan dibahas mengenai *Simulator Center of Gravity and Weight Management System (CG & WMS)* yang menggambarkan pengaturan posisi titik berat pesawat terbang akibat pemakaian bahan bakar.

Prinsip utama *Center of Gravity and Weight Management System (CG & WMS)* adalah mengatur beban (air sebagai beban) dalam *ballast system* sehingga posisi titik berat pesawat terbang akan sesuai harga yang telah ditentukan.

Prosesor yang digunakan pada simulator ini adalah Transputer T805 yang dilengkapi dengan Floating Point Unit (FPU) sebagai pengambil dan pengolah data dan Personal Computer (PC) 386DX sebagai *host* atau *downloader* sekaligus untuk menampilkan hasil, serta ditambah *serial link adapter* IMS C011-1 sebagai media komunikasi antara transputer dengan modul ADC.

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah Yang Maha Kuasa, yang telah melimpahkan rahmad dan hidayahNya, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan judul :

SIMULATOR CENTER OF GRAVITY AND WEIGHT MANAGEMENT SYSTEM (CG&WMS)
PADA PROGRAM UJI TERBANG DENGAN MENGGUNAKAN TRANSPUTER T805

Adapun Tugas Akhir ini mempunyai beban kredit 6 SKS (Satuan Kredit Semester) dan merupakan persyaratan guna memperoleh gelar kesarjanaan di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya.

Penulis tak lupa menyampaikan terima kasih yang sebesar-besarnya atas bimbingan, nasehat, dorongan serta fasilitas yang telah diberikan dari:

- Bapak Dr.Ir. Moch. Salehudin, M.Eng.Sc. selaku Ketua Jurusan Teknik Elektro FTI-ITS, Surabaya.
- Bapak Ir. Katjuk Astrowulan MSEE.
- Bapak Ir. Soetikno selaku Koordinator Bidang Studi Elektronika dan kepala Laboratorium R&D Elektronika sekaligus dosen pembimbing I.
- Bapak Ir. Hendra Kusuma selaku dosen pembimbing II.
- Bapak Ir. Iskandar Zulkarnain selaku dosen wali.
- Staf pengajar dan karyawan Jurusan Teknik Elektro, FTI-ITS, Surabaya.

- Ibu dan Bapak serta Mbak 'Rin dan Dik 'Yanti yang banyak membantu baik material maupun spiritual.
- Rekan-rekan bidang studi Elektronika, khususnya rekan di Laboratorium B203 dan semua pihak yang turut berperan dalam penyelesaian Tugas Akhir ini.

Akhir kata, penulis berharap semoga segala sesuatu yang telah dihasilkan dalam pelaksanaan Tugas Akhir ini dapat bermanfaat dan memperkaya khazanah ilmu pengetahuan serta memberikan kesejahteraan bagi umat manusia.

Surabaya, Juli 1994

Penulis

DAFTAR ISI

BAB	Halaman
JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika	3
1.7 Relevansi	4
2 TITIK BERAT DAN KESEIMBANGAN	5
2.1 Prinsip Titik Berat dan Keseimbangan	5

2.1.1	Penggeseran Beban Untuk Memindahkan Titik Berat . .	7
2.1.2	Perhitungan Posisi Titik Berat	8
2.2	Kestabilan	10
2.2.1	<i>CG Position</i> (Posisi Titik Berat)	11
2.2.2	Titik Berat Terlalu ke Depan	12
2.2.3	Titik Berat Terlalu ke Belakang	13
2.3	Berat dan Keseimbangan	13
2.3.1	Berat Pesawat Terbang	13
2.3.2	Keseimbangan Pesawat Terbang (<i>Airplane Balance</i>) . . .	14
2.4	Menentukan Titik Berat (<i>CG</i>)	14
2.5	Komponen Berat Pesawat Terbang yang Direncanakan	17
2.6	Perubahan Posisi Titik Berat Karena Pemakaian Bahan Bakar . .	19
2.7	Pemindahan Air Untuk Kompensasi Pemakaian Bahan Bakar . .	20
2.8	Pemindahan Air Untuk Kompensasi Perubahan Titik Berat . . .	22
3	TRANSPUTER	23
3.1	Sekilas Transputer	23
3.2	Keluarga Transputer	25
3.2.1	Transputer 16 bit	25
3.2.2	Transputer 32 bit	26
3.2.3	Transputer di Masa Datang	26
3.3	Arsitektur Transputer IMS T805	26
3.3.1	<i>Process Scheduller</i> (Penjadwal Proses)	29



3.3.2 Komunikasi Antar Proses	30
3.3.3 Interrupt	33
3.3.4 Memori	34
3.3.5 Pelayanan Sistem	36
3.4 Karakteristik Transputer IMS T805	36
3.4.1 Prosesor	38
3.4.2 <i>Floating Point Unit (FPU)</i>	39
3.4.3 Memori	40
3.4.4 Link	43
3.5 Aplikasi Transputer IMS T805	44
3.6 Konfigurasi	45
3.6.1 Prosesor dan Proses	45
3.6.2 Konfigurasi Transputer	46
3.7 <i>Serial Link Interface</i>	48
3.7.1 Adaptor Link	49
3.7.2 C011 Mode 1	49
4 PERENCANAAN HARDWARE	51
4.1 Perencanaan Hardware Sistem	51
4.2 Perencanaan Pengolah Sinyal Analog	51
4.2.1 Penguat Sample-Hold	52
4.2.2 Proteksi	54
4.2.3 Auto Range	55

4.2.3.1 Differential Amplifier	55
4.2.3.2 Komparator	55
4.2.3.3 Encoder Auto Range	56
4.2.3.4 Multiplexer	56
4.3 Tegangan Referensi	57
4.4 Rangkaian ADC	58
4.4.1 ADC 0804	58
4.4.2 Selector Range dan Data	59
4.5 Serial Interface	61
 5 PERENCANAAN SOFTWARE	 62
5.1 Transfer Function Sistem yang Direncanakan	63
5.2 Program Utama di Komputer (PC)	64
5.2.1 Menampilkan Simulasi	66
5.3 Program di Transputer	67
5.3.1 Mengambil Data ADC	68
5.3.2 Mengolah Data	69
5.4 Komunikasi Antara PC dan Transputer	70
 6 PENGUKURAN DAN PENGUJIAN	 72
6.1 Pengukuran Alat	72
6.1.1 Pengukuran Noise	72
6.1.2 Pengukuran Auto Range	73

	x
6.2 Pengukuran Modul ADC 0804	74
6.3 Serial Link IMS C011 Mode 1	76
6.4 Pengujian Sistem	77
7 PENUTUP	80
7.1 Kesimpulan	80
7.2 Saran	80
DAFTAR PUSTAKA	81
LAMPIRAN	

DAFTAR GAMBAR

GAMBAR	Halaman
2.1 Jungkat-jungkit Seimbang	5
2.2 Jungkat-jungkit Tidak Seimbang	6
2.3 Pemindahan Beban Agar Titik Berat Lurus dengan Titik Tumpu	8
2.4 Tanda Daerah Lengan (Jarak) di Samping Titik Tumpu	9
2.5 Kemungkinan Penentuan Posisi Datum (Titik Referensi)	10
2.6 Contoh Batasan Titik Berat yang Dibenarkan	12
2.7 Komponen Berat dan Jarak	16
2.8 Komponen Berat Pesawat Terbang yang Direncanakan	18
3.1 Blok Diagram Transputer	27
3.2 Register Transputer	29
3.3 Blok Diagram Transputer IMS T805	38
3.4 Peta Memori Transputer IMS T805	42
3.5 Prosesor dan Proses	46
3.6 Konfigurasi Satu Transputer	47
3.7 Konfigurasi Dua Transputer	47
3.8 Konfigurasi Jaringan Transputer	48
3.9 Blok Diagram C011 Mode 1	49
4.1 Blok Diagram Sistem	51
4.2 Blok Diagram Modul ADC	52
4.3 Rangkaian Sample Hold	53

4.4 Waktu Acquisasi dan Laju Drop Tegangan Terhadap Capasitor Hold .	53
4.5 Rangkaian Proteksi	54
4.6 Rangkaian Multiplexer	57
4.7 Rangkaian Tegangan Referensi	58
4.8 Time Diagram IMS C011 Mode 1	59
4.9 Rangkaian Selector Range dan Data	60
5.1 Flow Diagram Software Sistem	62
5.2 Transfer Function Sistem yang Direncanakan	63
5.3 Flowchart Program Utama	65
5.4 Flowchart Simulasi di PC	66
5.5 Flowchart Simulasi di Transputer	67
6.1 Pengukuran Noise	73
6.2 Batas CG_{INT} Pada Berbagai Jumlah Bahan Bakar	79

DAFTAR TABEL

TABEL	Halaman
2.1 Contoh Komponen Berat	15
2.2 Perhitungan Berat dan Momen Saat Take-off	15
3.1 Keluarga Transputer (Desember 1989)	25
3.2 Lokasi Memori	35
6.1 Hasil Pengukuran Auto Range	74
6.2 Hasil Konversi Modul ADC	77
6.3 Kondisi Take-off pada Berbagai Harga CG_{INIT}	78
6.4 Batas CG_{INIT} Pada Berbagai Jumlah W_F	79

BAB I

PENDAHULUAN

1.1 Latar Belakang

Center of Gravity (CG) atau titik berat pada pesawat terbang adalah salah satu parameter yang perlu diketahui. Karena posisi titik berat pesawat terbang dapat berubah akibat penggunaan bahan bakar, maka diperlukan adanya pengontrol untuk mengantisipasi perubahan posisi titik beratnya.

Prinsip utama *Center of Gravity and Weight Management System (CG & WMS)* adalah *ballast system* (sistem pengaturan beban) berfungsi mengantisipasi perubahan posisi titik berat pesawat terbang akibat perubahan berat pesawat (berkurangnya bahan bakar) selama beroperasi, dengan cara mengatur beban, dalam hal ini dipakai air sebagai beban.

Dalam simulator yang direncanakan, menggunakan Transputer T805 sebagai unit pemroses, PC-AT 386DX sebagai *host* dan sarana menampilkan hasil beserta serial link adapter Inmos C011 untuk komunikasi antara transputer dengan modul ADC.

1.2 Permasalahan

Permasalahan dari tugas akhir ini adalah bagaimana merencanakan dan membuat simulator sistem untuk menggambarkan pengendalian posisi titik berat pesawat terbang selama beroperasi akibat pemakaian bahan bakar.

1.3 Tujuan

Tujuan tugas akhir ini adalah membuat simulator yang menggambarkan pengendalian perubahan posisi titik berat pesawat terbang selama beroperasi karena perubahan berat pesawat akibat penggunaan bahan bakar dengan menggunakan transputer dan komputer.

1.4 Batasan Masalah

Topik utama dari tugas akhir ini dibatasi pada pembuatan simulator menggunakan transputer dan komputer yang mampu menggambarkan pengendalian perubahan posisi titik berat pesawat terbang selama beroperasi akibat perubahan berat pesawat dengan cara mengatur beban.

1.5 Metodologi

Untuk mencapai tujuan yang telah disebutkan di atas, dilakukan langkah-langkah sebagai berikut :

Langkah pertama adalah pemahaman mengenai pesawat yang akan dilakukan uji terbang tentang perubahan posisi titik beratnya akibat penggunaan bahan bakar. Mempelajari Transputer T805, rangkaian interface komunikasi serial dengan IMS C011 sebagai link adapter beserta software pendukungnya.

Langkah kedua perancangan simulator Center of Gravity and Weight Management System untuk mengendalikan perubahan posisi titik berat pesawat terbang akibat perubahan berat.



Langkah ketiga adalah pembuatan dan pengujian sistem untuk menghasilkan suatu kesimpulan mengenai kemampuan sistem yang telah dirancang. Akhirnya dari langkah-langkah tersebut di atas disusun dalam laporan tugas akhir.

1.6 Sistematika

Sistematika dari laporan tugas akhir terdiri dari VII bab yang tersusun sebagai berikut:

- Bab I adalah pendahuluan yang berisi tentang latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika, dan relevansi.
- Bab II adalah pembahasan mengenai dasar-dasar titik berat secara umum kemudian dilanjutkan secara khusus pada pesawat terbang.
- Bab III adalah pembahasan Transputer mengenai keluarga transputer, arsitektur transputer, karakteristik Transputer IMS T805, aplikasi transputer, konfigurasi transputer.
- Bab IV berisi perancangan hardware yang meliputi rangkaian signal condition, rangkaian ADC dan rangkaian serial.
- Bab V berisi pembahasan software yang digunakan untuk mengoperasikan sistem pada transputer dan komputer.
- Bab VI memuat pengujian dari sistem yang dirancang baik hardware maupun software.
- Bab VII adalah penutup yang berisi kesimpulan dan saran.

1.7 Relevansi

Perancangan Simulator Center of Gravity and Weight Management System (CG & WMS) ini diharapkan dapat memberikan sumbangan pikiran bagi industri pesawat terbang sebagai salah satu alternatif untuk dikembangkan lebih lanjut.

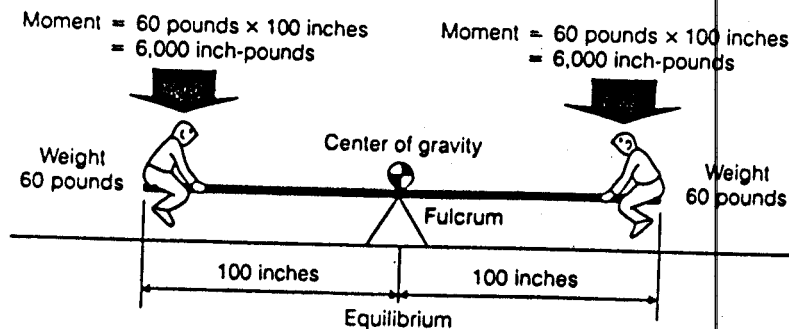
BAB II

TITIK BERAT DAN KESEIMBANGAN

2.1 Prinsip Titik Berat dan Keseimbangan

Terlihat pada gambar 2.1 dua anak yang sedang bermain jungkat-jungkit (*teeter-totter*). Berat badan mereka sama, dan jarak keduanya terhadap titik tumpu (*fulcrum*) juga sama. Kondisi seperti ini adalah seimbang sebab besar momen pada masing-masing ujung jungkat-jungkit sama. Keseimbangan tersebut akan terganggu jika salah satu dari mereka menghentakkan kakinya ketanah. Hubungan ini dapat digambarkan pada persamaan matematika:

$$\text{Momen} = \text{Berat} * \text{Jarak} \quad (2.1)$$



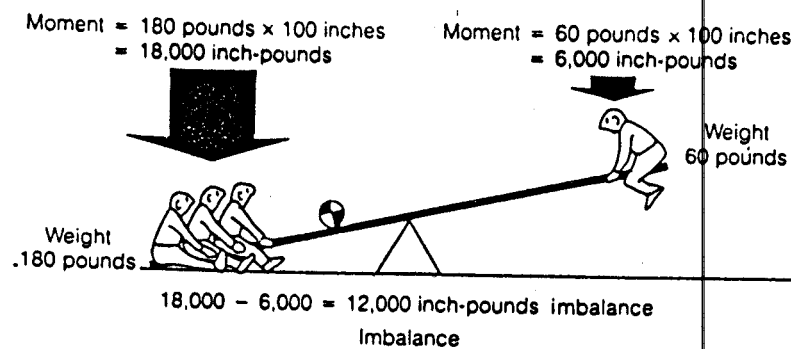
Gambar 2.1¹
Jungkat-jungkit Seimbang

Misalkan berat badan kedua anak tersebut 60 pound dan panjang jungkat-jungkit 200 inch. Titik tumpu di tengah, 100 inch dari ujung. Masing-masing akan

¹ Glaeser Dannis, *An Aviation to Fly*, Library of Congress Cataloging in Publication Data, 1985, hal.129.

mempunyai momen $100 \text{ inch} \times 60 \text{ pound}$ atau 6000 inch.pound . Keadaan tersebut adalah seimbang dan pada keadaan seimbang, titik berat merupakan titik dimana seluruh berat objek terkonsentrasi dan posisinya lurus dengan titik tumpu.

Apabila terjadi dua anak lain naik kesalah satu ujung jungkat-jungkit maka keseimbangan jadi kacau, dan salah satu ujungnya (dinaiki tiga anak) akan turun ketanah. Sebetulnya setelah dua anak naik, mereka menggeser posisi titik berat kearah mereka.



Gambar 2.2²
Jungkat-jungkit Tidak Seimbang

Gambar 2.2 di atas memperlihatkan bahwa berat tiga anak (masing-masing 60 pound) menghasilkan momen $180 \text{ pound} \times 100 \text{ inch}$ atau 18000 inch.pound . Momen ini jauh lebih besar dibanding 6000 inch.pound pada ujung lain.

Ada dua cara membuat jungkat-jungkit seimbang lagi. Salah satu caranya mengembalikan kondisi seimbang adalah dengan menggeser posisi ketiga anak tersebut kearah titik tumpu sampai momen yang dihasilkan sama dengan di ujung yang lain.

² Ibid, hal.130.

2.1.1 Penggeseran Beban Untuk Memindahkan Titik Berat

Jika menggeser posisi ketiga anak tersebut sampai jungkat-jungkit seimbang, masalahnya adalah dimana tepatnya mereka harus digeser. Hal ini dapat diselesaikan dengan meninjau persamaan (2.1).

$$\begin{aligned} \text{Momen} &= \text{Berat} * \text{Jarak} \\ \text{Jarak} &= \frac{\text{Momen}}{\text{Berat}} \end{aligned} \quad (2.2)$$

Untuk menyeimbangkan momen 6000 inch.pound disalah satu ujung dengan berat 180 pound di ujung yang lain dapat dihitung sebagai berikut:

$$\frac{6000 \text{ inch.pound}}{180 \text{ pound}} = 33,33 \text{ inch}$$

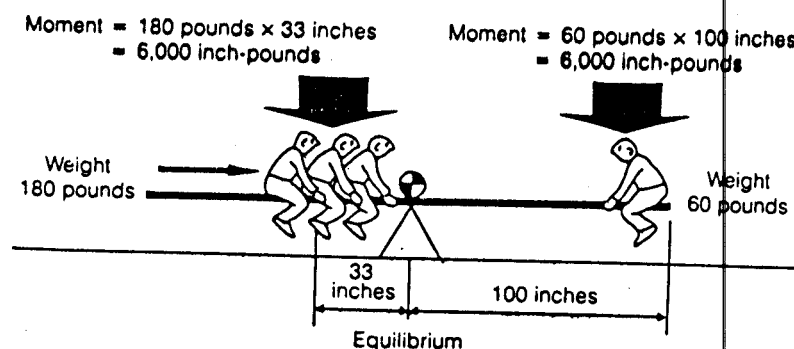
Ketiga anak tersebut harus dipindahkan ke titik 33,33 inch dari titik tumpu.

Check momen pada kondisi baru:

$$180 \text{ pound} * 33,33 \text{ inch} = 6000 \text{ inch.pound}$$

$$60 \text{ pound} * 100 \text{ inch} = 6000 \text{ inch.pound}$$

Didapat momen yang sama, maka keadaannya seimbang dan titik beratnya lurus dengan titik tumpu.



Gambar 2.3³
Pemindahan Beban Agar Titik Berat Lurus dengan Titik Tumpu

2.1.2 Perhitungan Posisi Titik Berat

Cara kedua membuat jungkat-jungkit seimbang kembali dengan jalan memindahkan tumpuan. Sebelumnya harus menentukan semua jarak beban dari titik tumpu. Karena itu ditentukan daerah disebelah kiri titik tumpu diberi tanda minus dan daerah di sebelah kanan plus. Tanda minus dan plus bebas tergantung penentuan dan tidak akan merubah hasil perhitungan.

Prosedur perhitungan titik berat dimulai dari:

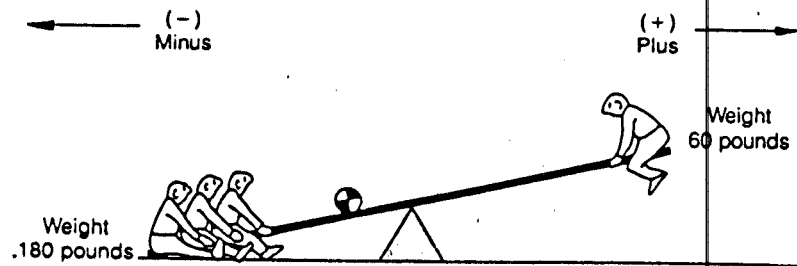
1. Menjumlah semua berat untuk mendapatkan berat total.
2. Menghitung semua momen dan menjumlahkannya untuk mendapatkan momen total.
3. Membagi momen total dengan berat total, dengan menggunakan rumus

$$CG \text{ inch} = \frac{\text{Momen total inch.pound}}{\text{Berat total pound}} \quad (2.3)$$

Hasil di atas adalah lokasi titik berat (CG) terhadap titik tumpu, dapat

³ Ibid, hal.131.

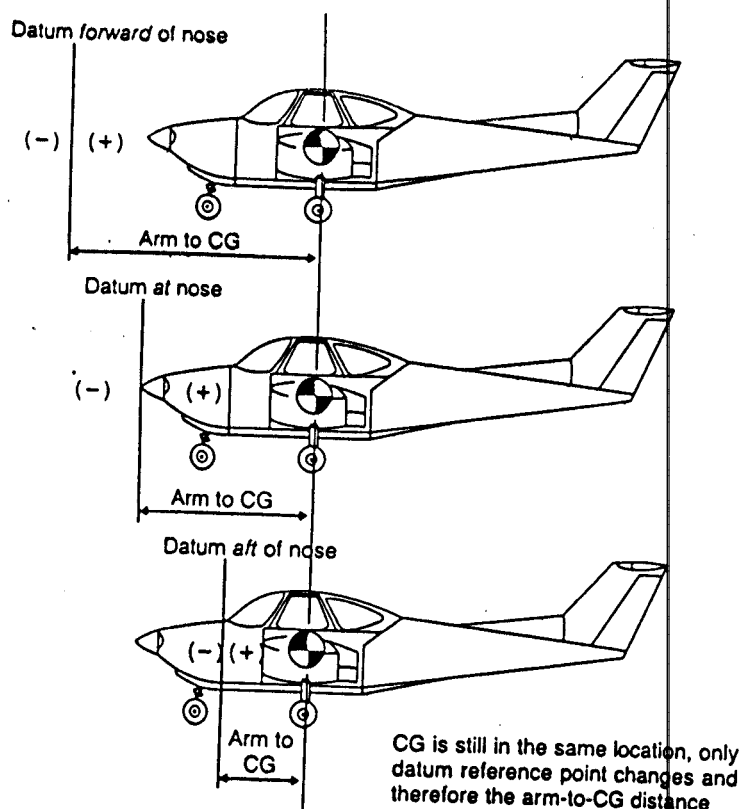
dilihat pada gambar berikut.



Gambar 2.4⁴
Tanda Daerah Lengan (Jarak) di Samping Titik Tumpu

Pada pesawat terbang penentuan *datum* (titik referensi) dalam pengukuran jarak ada beberapa kemungkinan seperti pada gambar 2.5. Untuk selanjutnya ditentukan lokasi di belakang atau di sebelah kanan datum bertanda plus, dan di depan atau di sebelah kiri datum bertanda minus. Dari ketiga kemungkinan tersebut dalam perhitungan tidak akan merubah posisi titik berat.

⁴ Loc Cit.



Gambar 2.5⁵
Kemungkinan Penentuan Posisi Datum (Titik Referensi)

2.2 Kestabilan

Pesawat terbang mempunyai tiga sumbu rotasi dalam gerakannya. Keseimbangan memanjang (*longitudinal stability*) mempengaruhi gerakan naik atau turun, atau kecenderungan pesawat terbang terhadap salah satu sisi.

Pesawat terbang yang mempunyai kestabilan memanjang akan cenderung kembali pada posisi yang telah diatur setelah terjadi perubahan. Hal ini memberi keuntungan karena pesawat terbang yang bersifat seperti ini cenderung melawan

⁵ Ibid, hal.133.

perubahan naik / turunnya hidung pesawat terbang.

Jika pesawat terbang secara memanjang tidak stabil, akan cenderung naik atau turun sampai kehilangan kecepatan terbang atau menghunjam. Sehingga keseimbangan memanjang yang tidak stabil pada pesawat terbang sangat berbahaya untuk penerbangan.

Posisi titik angkat (*Center of Lift*) yang berada pada garis sepanjang sayap, selama terbang akan mempengaruhi keadaan atau sifat terbang. Apabila berubah ke depan sudut terbang akan bertambah dan jika ke belakang sudut terbang berkurang. Jadi keadaan terbang cenderung dipengaruhi oleh posisi titik angkat relatif terhadap perubahan titik berat.

Misalnya dengan terbang naik atau sudut tinggi dan titik angkat di depan titik berat maka kecenderungan hidung pesawat turun akan berkurang. Posisi titik berat relatif terhadap titik angkat adalah faktor penting kestabilan memanjang. Jika titik berat jauh ke depan, pesawat terbang terbebani ke depan (terbebani ke hidung pesawat terbang) dan jika titik berat ke belakang pesawat terbang akan terbebani pada ekornya.

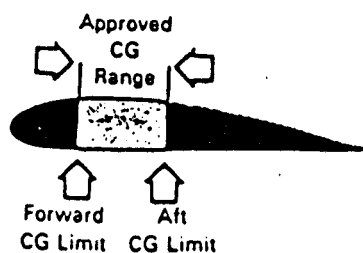
2.2.1 CG Position (Posisi Titik Berat)

Posisi titik berat merupakan kunci utama stabilitas memanjang. Pembagian berat dalam pesawat terbang dihitung berdasarkan posisi titik beratnya. Berat ini termasuk berat dasar (*basic weight*) dari pesawat terbang itu sendiri demikian pula berat bahan bakar.

Sejak menentukan berat pesawat terbang dan pembagian berat biasanya

merupakan hal yang sulit, sebab kita harus menghitung dan menentukan titik beratnya. Meletakkan apa dalam pesawat terbang dan dimana tempatnya.

Kita boleh berharap, untuk pesawat terbang yang dapat dikontrol selama terbang, titik berat harus berada dalam jarak antara batas depan (*Forward Center of Gravity*) dan batas belakang (*Aft Center of Gravity*). Semua pesawat terbang memiliki batas depan dan batas belakang untuk posisi titik beratnya.



Gambar 2.6⁶
Contoh Batasan Titik Berat yang Dibenarkan

Ketika titik berat dalam daerah titik berat yang dibenarkan, pesawat terbang tidak hanya dapat dikontrol, tetapi juga mempunyai kestabilan memanjang yang baik. Jika titik berat berada sekitar batas depan atau batas belakang yang dibenarkan untuk daerah titik berat, tidak boleh meremehkan kestabilan memanjang dan harus diperhatikan. Dimana keberhasilan *elevator* apa masih memadai untuk mengontrol pesawat guna keberhasilan *manuver*nya.

2.2.2 Titik Berat Terlalu ke Depan

Jika beban pesawat terbang yang mengakibatkan titik berat berada di depan

⁶ Sanderson Jeppessen, *Aviation Fundamental*, 1988, hal.1.47.

batas depan titik berat, hal ini menjadikan hidung pesawat terbang sangat berat. Keadaan ini menjadikan perpindahan titik berat ke depan yang lebih buruk yaitu ke posisi ekstrim depan. Kenyataan ini membuat pesawat terbang terlalu stabil. Kemungkinan, keberhasilan mengangkat hidung pesawat terbang sangat kecil.

2.2.3 Titik Berat Terlalu ke Belakang

Titik berat teletak di belakang batas yang dibenarkan lebih berbahaya dibandingkan titik berat terlalu ke depan. Dengan titik berat terlalu ke belakang pesawat terbang akan terbebani pada ekornya dan sangat tidak stabil dalam penerbangan untuk segala kecepatan.

2.3 Berat dan Keseimbangan

Didepan telah dibahas mengenai lokasi atau posisi titik berat pesawat terbang yang berhubungan dengan kestabilan. Pada bagian ini lebih jauh tentang konsep titik berat dalam teknik perhitungan berat dan keseimbangan.

2.3.1 Berat Pesawat Terbang

Sebagian besar pabrik menggunakan istilah *Basic Empty Weight*, untuk berat standard pesawat terbang, tambahan perlengkapan, minyak mesin, tanpa bahan bakar. Sebelumnya menggunakan istilah *Lincensed Empty Weight*, biasanya sama dengan *Basic Empty Weight* kecuali tanpa minyak mesin, dan hanya termasuk berat salurannya saja.

Standard berat yang digunakan untuk perhitungan berat bahan bakar

penerbangan (*aviation gasolin / avgas*) dan minyak mesin. Avgas beratnya 6 pound per gallon, sementara berat minyak mesin 7,5 pound per gallon. Kapasitas minyak mesin biasanya menggunakan istilah dalam volume, karena itu kita harus mengkonversikan ke berat.

2.3.2 Keseimbangan Pesawat Terbang (*Airplane Balance*)

Titik berat pada pesawat terbang didefinisikan titik dimana semua keseimbangan berat pesawat terbang terkonsentrasi. Posisi titik ini sangat penting untuk kestabilan pesawat terbang. Mengabaikan keseimbangan beban pesawat terbang bisa mengakibatkan masalah kontrol yang serius. Tetapi hal ini dapat diatasi dengan menghitung dan menempatkan letak prioritas titik berat untuk terbang dan kemudian membandingkan dengan letak hasil perhitungan yang memenuhi persyaratan dari pabrik.

Jika letaknya di luar batas yang dibenarkan, dapat diatasi dengan merubah letak beban yang ada misalnya bahan bakar, penumpang, muatan, sehingga dapat memindahkan letak titik berat ke batasan yang dibenarkan. Hal ini sama dengan kepentingan memperbaiki pembagian berat selama terbang.

2.4 Menentukan Titik Berat

Secara umum metode yang dapat digunakan untuk menentukan titik berat (CG) ada 3 metode. Ketiga metode tersebut adalah metode perhitungan, metode tabel dan metode grafik. Untuk selanjutnya dalam tugas akhir ini hanya dibahas dan menggunakan metode perhitungan.

Tabel 2.1 ⁷
Contoh Komponen Berat

	Berat (pound)
Berat dasar (basic empty weight)	1600
Pilot dan penumpang di depan	340
Penumpang di belakang	340
Bagasi	35
Bahan bakar	330

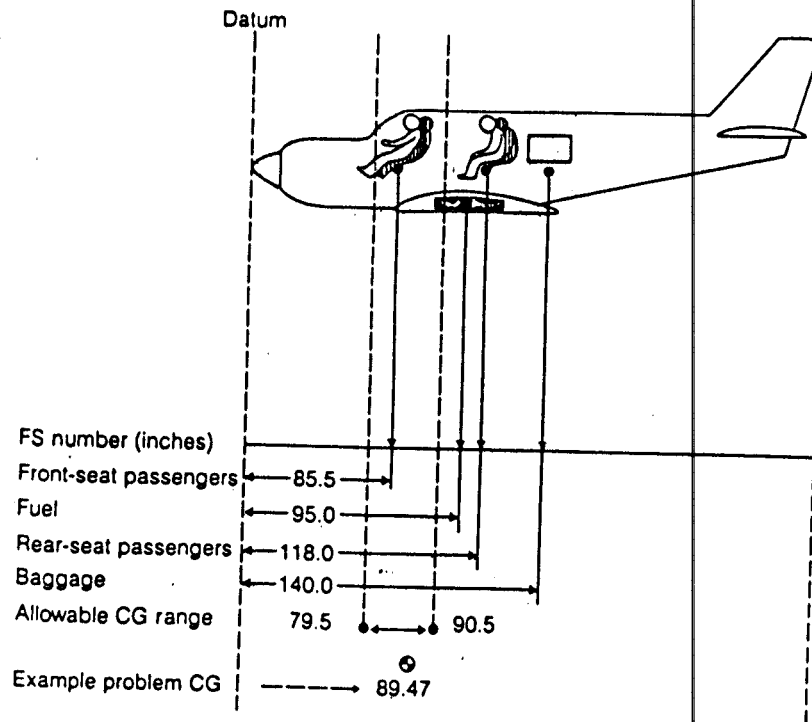
Misalkan terdapat data seperti pada tabel 2.1, yang memuat informasi berat pesawat terbang beserta muatannya. Guna mendapatkan informasi yang lebih lengkap harus dicari data jarak atau lengan dari masing-masing komponen berat diukur relatif terhadap datum. Kemudian mengalikan antara masing-masing komponen berat dengan jaraknya untuk mendapatkan besar momen, seperti pada tabel 2.2.

Tabel 2.2 ⁸
Perhitungan Berat dan Momen Saat Takeoff

	Berat (pound)	Jarak (inchi)	Momen (pound.inch)
Berat dasar (basic empty weight)	1600	82,0	131,200
Pilot dan penumpang di depan	340	85,5	29,070
Penumpang di belakang	340	118,00	40,120
Bagasi	35	140,0	4,9000
Bahan bakar	330	95,0	31,350
Total	2645		236,640

⁷ Glaeser Dennis, Op Cit, hal.135.

⁸ Ibid, hal.136.



Gambar 2.7⁹
Komponen Berat dan Jarak

Setelah didapat data komponen berat, jarak dan momen dilanjutkan dengan menentukan titik beratnya dengan menggunakan persamaan (2.3).

Akhirnya didapat :

$$CG_{TakeOff} = \frac{\sum \text{Momen}}{\sum \text{Berat}}$$

$$CG_{TO} = \frac{236,640 \text{ inch.pound}}{2645 \text{ pound}}$$

$$CG_{TO} = 89,47 \text{ inch}$$

⁹ Ibid, hal.135.

Terlihat bahwa lokasi CG pada saat *take off* (tinggal landas) masih terdapat dalam batas yang dibenarkan.

Selanjutnya di check titik berat pesawat terbang pada saat *landing* (mendarat) apakah masih aman. Misalkan selama beroperasi dibutuhkan 45 gallon atau 270 pound (1_gallon = 6_pound) bahan bakar, maka perlu dihitung kembali posisi titik beratnya. Mengurangkan berat dan momen selama beroperasi terhadap kondisi take-off, setelah dihitung didapat.

$$CG_{Landing} = \frac{(236640 - 25650) \text{ inch.pound}}{(2645 - 270) \text{ pound}}$$

$$CG_L = \frac{210990 \text{ inch.pound}}{2375 \text{ pound}}$$

$$CG_L = 88.84 \text{ inch}$$

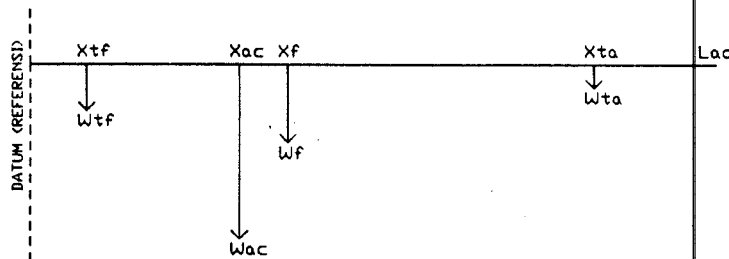
Titik berat berpindah atau bergeser sedikit dalam batasan yang dibenarkan. Jadi dapat disimpulkan bahwa kondisi di atas aman untuk penerbangan.

2.5 Komponen Berat Pesawat Terbang yang Direncanakan

Seperti telah dijelaskan didepan bahwa dalam tugas akhir ini direncanakan dibuat simulator yang menggambarkan pengaturan posisi titik berat pesawat terbang selama beroperasi yang berubah akibat pemakaian bahan bakar.

Parameter yang direncanakan dalam simulator ini terdiri dari komponen berat dan jaraknya terhadap datum yaitu berat pesawat terbang (W_{AC}), berat bahan bakar (W_F), berat air di tangki depan (W_{TF}) serta berat air di tangki belakang (W_{TA}), seperti pada gambar 2.8. Untuk selanjutnya semua perhitungan didasarkan

pada parameter-parameter tersebut dengan mengabaikan parameter lainnya.



Gambar 2.8
Komponen Berat Pesawat Terbang yang Direncanakan

- X_{TF} : Jarak (*Arm*) titik berat tangki air depan (*Forward*).
- X_F : Jarak titik berat bahan bakar (*Fuel*).
- X_{AC} : Jarak titik berat pesawat terbang.
- X_{TA} : Jarak titik berat tangki air belakang (*Aft*).
- W_{TF} : Berat (*Weight*) tangki air depan.
- W_F : Berat bahan bakar.
- W_{AC} : Berat pesawat terbang.
- W_{TA} : Berat tangki air belakang.
- L_{AC} : Panjang pesawat terbang.

Ujung pesawat terbang digunakan sebagai datum atau titik referensi sehingga mempunyai jarak nol, dan semua jarak diukur relatif terhadap titik ini.

Jarak titik berat dapat dihitung berdasarkan perbandingan antara moment dengan berat. Sedangkan moment merupakan hasil perkalian antara jarak dan berat. Sehingga posisi atau jarak titik berat (X_{CG}) dapat dihitung sebagai berikut.

Dimana :

M_{TO} : Moment pesawat terbang saat take off.

W_{TO} : Berat pesawat terbang saat take off.



$$X_{CG} = \frac{(W_{TF} * X_{TF}) + (W_{AC} * X_{AC}) + (W_F * X_F) + (W_{TA} * X_{TA})}{(W_{TF} + W_{AC} + W_F + W_{TA})} \quad (2.4)$$
$$X_{CG} = \frac{M_{TO}}{W_{TO}}$$

Kondisi diatas pada waktu pesawat terbang *Take Off* (tinggal landas).

Berdasarkan panjang pesawat terbang dari (2.4) didapat:

$$CG = \frac{X_{CG}}{L_{AC}} * 100\% \quad MAC \quad (2.5)$$

$$X_{CG} = \frac{CG * L_{AC}}{100} \quad (2.6)$$

Pada saat beroperasi pesawat terbang selalu membutuhkan bahan bakar, sehingga berat bahan bakar akan berubah atau berkurang. Dengan berkurangnya berat bahan bakar besar momen-pun akan berubah, begitu juga posisi titik beratnya. Dengan kata lain perubahan berat bahan bakar akan mempengaruhi posisi titik berat pesawat terbang.

2.6 Perubahan Posisi Titik Berat Karena Penggunaan Bahan Bakar

Misalkan W_{FU} adalan bahan bakar yang digunakan, sisa bahan bakarnya ($W_{FU} - W_{FU}$) maka jarak titik beratnya adalah:

$$X_{CG_{FU}} = \frac{(W_{TF} * X_{TF}) + (W_{AC} * X_{AC}) + [(W_F - W_{FU}) * X_{FU}] + (W_{TA} * X_{TA})}{(W_{TF} + W_{AC} + W_F - W_{FU} + W_{TA})} \quad (2.7)$$

$$X_{CG_{FU}} = \frac{M_{T0} - (X_F * W_{FU})}{W_{T0} - W_{FU}}$$

seperti pada persamaaan (2.6) dari (2.7) didapat:

$$CG_{FU} = \frac{X_{CG_{FU}}}{L_{AC}} * 100\% \quad Mac \quad (2.8)$$

$$X_{CG_{FU}} = \frac{CG_{FU} * L_{AC}}{100} \quad (2.9)$$

Mengurangkan persamaan (2.7) dari persamaan (2.9) didapat:

$$X_{CG_{FU}} - X_{CG} = \frac{L_{AC}}{100} * (CG_{FU} - CG) \quad (2.10)$$

$$CG_{FU} - CG = (X_{CG_{FU}} - X_{CG}) * \frac{100}{L_{AC}}$$

$$CG_{FU} - CG = \frac{W_{FU} * [(CG * L_{AC}) - (100 * X_F)]}{L_{AC} * (W_{T0} - W_{FU})} \quad (2.11)$$

$$CG_{FU} - CG = \frac{(M_{T0} * W_{FU}) - (W_{T0} * X_F * W_{FU})}{W_{T0} (W_{T0} - W_{FU})} * \frac{100}{L_{AC}} \quad (2.12)$$

2.7 Perpindahan Air Untuk Kompensasi Penggunaan Bahan Bakar

Untuk mengembalikan ke posisi awal (sesuai setting) harus memindahkan beban (air) dari tangki air belakang ke tangki air depan atau sebaliknya,

tergantung arah pergeseran titik berat. Jika posisi titik berat berkurang perpindahan air dari tangki depan ke belakang dan begitu pula sebaliknya apabila posisi titik berat bertambah perpindahan air dari tangki belakang ke depan. Misalkan W_{MF} adalah berat air yang dipindahkan.

$$X_{CG_{WMP}} = \frac{X_{TF}*(W_{TF}-W_{MF})+X_F*(W_F-W_{FU})+(X_{AC}*W_{AC})+X_{TA}*(W_{TA}+W_{MF})}{(W_{TF}+W_{MF}) + (W_F-W_{FU}) + W_{AC} + (W_{TA}+W_{MF})}$$

$$X_{CG_{WMP}} = \frac{X_{TF}*(W_{TF}-W_{MF})+X_F*(W_F-W_{FU})+(X_{AC}*W_{AC})+X_{TA}*(W_{TA}+W_{MF})}{W_{TO} - W_{FU}}$$
(2.13)

Untuk kompensasi, titik berat harus tetap atau sama maka :

$$X_{CG_{WMP}} = X_{CG}$$
(2.14)

Substitusi persamaan (2.4) dan (2.12) ke (2.13) :

$$W_{FU} * M_{TO} = (X_{TF}*W_{MF}*W_{TO})+(X_F*W_{FU}*W_{TO})-(X_{TA}*W_{MF}*W_{TO})$$
(2.15)

$$W_{FU} * (X_{CG}-X_F) = W_{MF} * (X_{TF}-X_{TA})$$

Dalam bentuk lain dapat ditulis :

$$W_{MF} = W_{FU} * \frac{(X_{CG}-X_F)}{(X_{TF}-X_{TA})}$$

$$W_{MF} = W_{FU} * \frac{[CG*L_{AC} - 100*X_F]}{100 * (X_{TF}-X_{TA})}$$
(2.16)

Jadi W_{MF} ini adalah berat air yang harus dipindahkan untuk mengkompensasi perubahan berat bahan bakar.

2.8 Perpindahan Air Untuk Kompensasi Perubahan Titik Berat

Selama terbang dilakukan pemindahan air dari salah satu tangki ke tangki lain akibat perubahan berat, pemindahan air ini akan mempengaruhi posisi titik berat pesawat terbang. Berdasarkan perubahan posisi ini dapat dicari berat air untuk mengkompensasi perubahannya dengan mencari selisih antara CG dengan CGrequest.

Berikut ini untuk menghitung berat air yang harus dipindahkan karena adanya perbedaan antara CG dengan CGrequest. Misalkan W_{MR} adalah berat air yang harus dipindahkan.

$$X_{CG_R} = \frac{X_{TF}(W_{TF} - W_{MF} - W_{MR}) + X_F(W_F - W_{FU}) + X_{AC}W_{AC} + X_{TA}(W_{TA} + W_{MF} + W_{MR})}{W_{TO} - W_{FU}} \quad (2.17)$$

Hasil pengurangan persamaan (2.13) pada persamaan (2.17).

$$X_{CG_{WMF}} - X_{CG_R} = \frac{X_{TA} - X_{TF}}{W_{TO} - W_{FU}} * W_{MR} \quad (2.18)$$

$$X_{CG_R} = \frac{CG_R * L_{AC}}{100} \quad (2.19)$$

$$X_{CG_{WMF}} = \frac{CG_{WMF} * L_{AC}}{100}$$

Dari selisih diantara persamaan (2.18)

$$W_{MR} = \frac{L_{AC} * (W_{TO} - W_{FU})}{100 * (X_{TA} - X_{TF})} * (CG_R - CG_{WMF}) \quad (2.20)$$

Jadi W_{MR} ini adalah berat air yang harus dipindahkan untuk mengkompensasi perubahan posisi titik berat pesawat terbang.

BAB III

TRANSPUTER

3.1 Sekilas Transputer

Penemuan ilmiah dan teknologi komputer hampir tak dapat dipisahkan, meskipun kemampuan komputer yang ada saat ini sangat mengagumkan, tetapi kebutuhan untuk komputasi yang lebih cepat lagi terus dikembangkan, ini disebabkan persoalan-persoalan utama dari para pakar yang diselesaikan oleh komputer sudah sangat kompleks, bahkan sudah melebihi kemampuan dari superkomputer yang ada saat ini yang mana kemampuan eksekusinya 1.5 milyar kalkulasi per detik.

Para pakar berusaha untuk mengembangkan kecepatan komputer tetapi mereka menemui keterbatasan terhadap sifat fisik semikonduktor yaitu panjang gelombang cahaya yang digunakan untuk penggambaran sirkuit ke wafer, disipasi panas akibat hambatan arus pada konduktor dan kebocoran elektron pada lapisan insulasi.

Salah satu upaya mengatasi masalah kecepatan yaitu penemuan sirkuit *photonic* yang menggunakan cahaya kilat, termasuk arus listrik, untuk mentransformasikan informasi digital melalui serat optik. Serat optik ini hampir tanpa disipasi panas dan tidak membutuhkan insulasi elektris. Konsep yang lebih radikal dari sirkuit photonic adalah *biochip*, yaitu komponen tiga dimensi yang dibuat dari molekul-molekul organik. Molekul-molekul ini dapat dipaketkan bersama-sama dalam densitas yang lebih besar dari komponen semikonduktor.

Pengembangan ke arah tersebut sangat menarik tetapi tidak dapat direalisasikan untuk pemakaian komersial saat ini, juga memerlukan biaya yang sangat tinggi untuk tiap unit dan sistem. Superkomputer menggunakan Gallium Arsenide atau supercooled sirkuit CMOS yang membutuhkan memori yang sangat cepat dan trilyun rupiah untuk memiliki serta mengoperasikan.

Berdasarkan kendala-kendala diatas, multiproses adalah salah satu alternatif yang paling mungkin. Persoalan-persoalan dipecahkan dengan menggunakan lusinan bahkan ratusan komputer yang bekerja secara paralel secara bersama-sama dalam suatu jaringan. Strategi ini meniru sel-sel otak (neuron), yang terikat bersama dalam jaringan yang dapat mengenali pola secara cepat atau mengolah informasi.

Diilhami oleh cara kerja sel-sel otak maka diciptakan suatu sistem yang dapat bekerja paralel. *Transputer* adalah prosesor yang didesain untuk dapat bekerja secara paralel dengan satu atau lebih transputer lainnya. Transputer dapat meningkatkan kecepatannya dengan mengeksekusi bagian-bagian dari sebuah persoalan secara bersama-sama. Sebuah transputer menangani bagian yang berbeda dari sebuah persoalan dengan cara mengeksekusi tiap bagian secara bergantian dengan cepat atau sebuah jaringan dari transputer menangani bagian-bagian tersebut secara simultan dengan mendistribusikannya ke tiap-tiap transputer. Untuk membagi suatu masalah menjadi bagian-bagian yang akan diproses oleh masing-masing transputer secara efisien dan efektif diperlukan pembagian beban kerja yang seimbang pada tiap transputer (*load balancing*).

3.2 Keluarga Transputer

Keluarga transputer terdiri dari tiga kelompok utama : seri T2 16 bit, seri T4 32 bit (hanya integer) dan seri T8 32 bit dimana didalamnya terdapat *Floating Point Unit (FPU)*.

Tabel 3.1 ¹⁰
Keluarga Transputer (Desember 1989)

	Transputer 16 bit			
	T212	T222	T225	M212
Panjang word	16	16	16	16
Internal RAM	2K	4K	4K	2K
Jumlah Link	4	4	4	2
Instruksi lanjut	n	y	y	n
Instruksi Debug	n	n	y	n

	Transputer 32 bit					
	T400	T414	T425	T800	T801	T805
Panjang word	32	32	32	32	32	32
Internal RAM	2K	2K	4K	4K	4K	4K
Jumlah Link	2	4	4	4	4	4
Hardware FPU	n	n	n	y	y	y

3.2.1 Transputer 16 bit

T212 adalah dasar transputer 16 bit, dengan 2Kbyte internal (*on-chip*) statik memori, T222 dilengkapi internal statik memori dan instruksi lanjut. T225 hampir sama dengan T222 tetapi berisi instruksi yang mendukung *debugging*. M212 adalah turunan T212 dilengkapi *disk interface*, 1Kbyte RAM dan 4Kbyte internal ROM berisi perangkat pengontrol disk. M212 mempunyai hanya dua *external link*, dua yang lain digunakan untuk *disk interface*.

¹⁰ Graham Ian, *The Transputer Handbook*, Printece Hall International (UK) Ltd, Great Britain, 1994, hal.16.

3.2.2 Transputer 32 bit

T414 sebagai dasar keluarga transputer 32 bit, mempunyai prosesor integer 32 bit dengan 2 Kbyte internal statik memori. T425 versi lebih lanjut dari T414, dengan instruksi pemindahan blok. Prosesor ini dilengkapi instruksi tambahan guna mendukung *debugging*, dan dikehendaki untuk mengganti T414. Sedangkan T400 lebih sederhana dan lebih murah dari versi T425, hanya 2 link dan 2 Kbyte internal memori.

Transputer pertama yang didesain khusus untuk perhitungan numerik adalah T800. Mempunyai 4 Kbyte internal memori, dan Floating Point Unit (FPU) standard IEEE 754 khusus bilangan 32 bit dan 64 bit. FPU dalam chip akan memberikan unjuk kerja floating point yang baik. T805 adalah pengembangan T800 dengan pendukung *debugging* hal ini sama dengan instruksi pada T225 dan T425. T801 tidak jauh beda tetapi mempunyai external memori RAM lebih cepat.

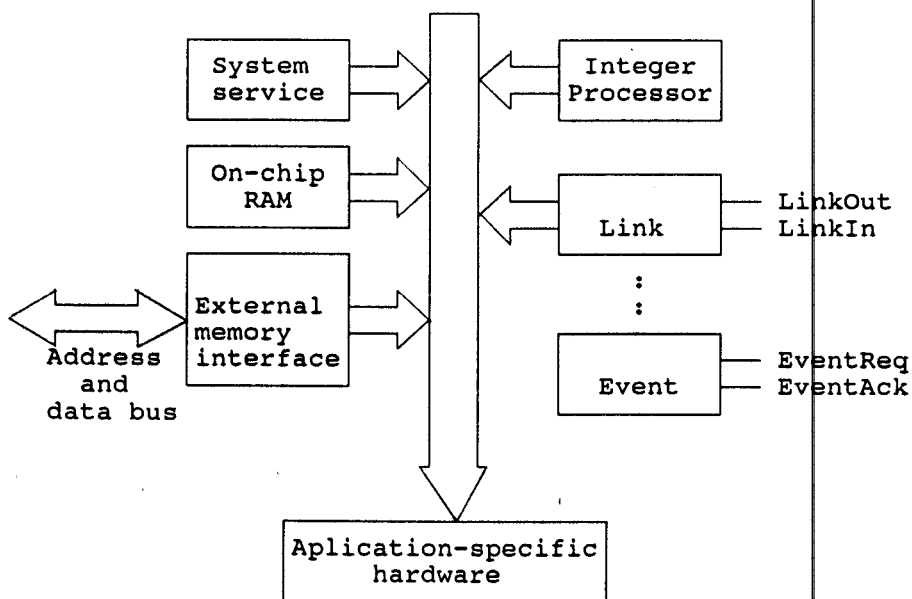
3.2.3 Transputer di Masa Datang

INMOS baru-baru ini mengumumkan rencana mereka untuk pengembangan transputer berkecepatan lebih tinggi, dengan kode H1. Rencananya prosesor ini kira-kira sepuluh kali kecepatan transputer sekarang, dan link memungkinkan transfer data 80 Mbyte perdetik.

3.3 Arsitektur Transputer IMS T805

Arsitektur Transputer mendefinisikan sebuah keluarga dari komponen VLSI yang dapat diprogram. Definisi dari arsitektur secara alamiah menjelaskan

aspek-aspek logikal yang mendefinisikan bagaimana mendesain dan memprogram sebuah sistem interkoneksi transputer-transputer, dan aspek fisik yang mendefinisikan bagaimana transputer sebagai komponen VLSI diinterkoneksi dan dikontrol.



Gambar 3.1¹¹
Blok Diagram Transputer

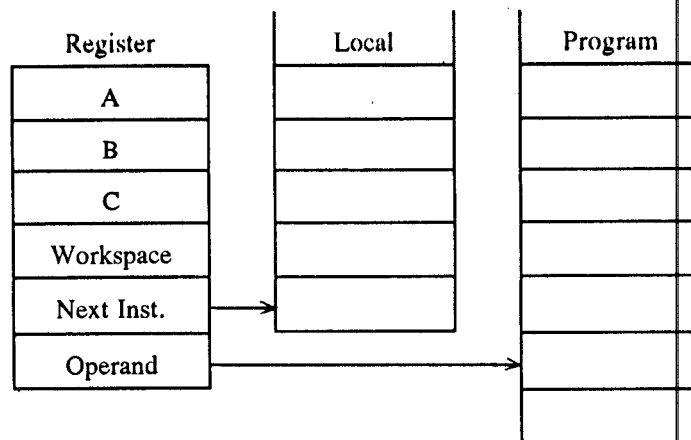
Biasanya anggota keluarga dari transputer adalah sebuah chip tunggal yang terdiri dari prosesor utama, memori dan link-link komunikasi yang memungkinkan hubungan antara transputer. Sebuah transputer dapat digunakan dalam sebuah sistem prosesor tunggal atau dalam jaringan untuk membangun sistem *concurrent* berkemampuan tinggi. Sebuah jaringan dari transputer dan kontrol peripheral sangat mudah disusun dengan menggunakan komunikasi titik per titik.

¹¹ Ibid, hal.8.

Rancangan internal transputer berbeda dengan prosesor-prosesor pada umumnya. Konsep dasar transputer adalah proses dan ini dapat dilihat pada set-set instruksinya. Sebuah proses menggambarkan sebuah individu yang mempunyai kerja tertentu dan transputer men-switch antar proses yang sedang berjalan untuk mendapatkan efek bahwa semua proses tersebut berjalan paralel. Proses switching ini biasa disebut *multitasking* dan biasanya ditangani oleh sistem operasi, tetapi pada transputer semuanya diimplementasikan secara hardware.

Model register dari transputer seperti ditunjukkan pada gambar 3.2. Semua register adalah 16 atau 32 bits, tergantung pada panjang word dari transputer tersebut. Register A, B dan C membentuk sebuah stack evaluasi dan instruksi-instruksi transputer didesain seputar penggunaan dari stack ini daripada penggunaan register *general purpose*. Dengan tiga register stack mendapatkan hasil yang baik dalam kemampuan mengevaluasi ekspresi-ekspresi pada stack tersebut dan menjadi lebih sedikit dalam menyimpan ke memori saat terjadi switch pada suatu proses.

Register W adalah *workspace pointer*, sebuah pointer dalam variabel lokal diasosiasikan dengan pelaksanaan proses saat itu. Banyak instruksi yang menunjuk ke data melalui offsetnya dari workspace pointer. Register penunjuk instruksi menunjuk I poin ke instruksi berikutnya yang akan dijalankan, dan ini analogi dengan program counter (PC) pada prosesor-prosesor konvensional. Register operan O digunakan dalam penyusunan dari operan-operan instruksi.



Gambar 3.2 ¹²
Register Transputer

3.3.1 *Process Scheduller* (Penjadwal Proses)

Salah satu kelebihan dari arsitektur transputer adalah adanya sebuah process scheduller terkode-mikro yang mengatur dua antrian proses, satu pada prioritas tinggi dan satu pada prioritas rendah. Proses-proses dalam antrian prioritas tinggi diijinkan untuk dieksekusi sampai selesai, atau saat membutuhkan input/output dari suatu channel. Tetapi proses-proses prioritas rendah tetap dieksekusi secara *timeslice* pada interval sekitar 1 ms, dan dapat diinterupsi oleh sebuah proses prioritas tinggi.

Antrian diimplementasikan sebagai *linked list* melalui workspace dari proses aktif, awal dan akhir antrian ditunjukkan oleh dua pasangan register, satu untuk setiap prioritas. Diasosiasikan dengan setiap level prioritas dari sebuah timer dan antrian sebuah proses timer. *Timer tick* proses prioritas tinggi setiap 1 mikrodetik, dan untuk proses prioritas rendah setiap 64 mikrodetik, tepatnya

¹² _____, *Transputer Architecture and Overview*, Computer System Architects (CSA), USA, 1990, hal.28.

15,625 tick setiap detik.

Jumlah bit dalam register timer tergantung pada model prosesor, untuk 16 bit prosesor timer prioritas tinggi akan berputar kira-kira 65,5 milidetik, untuk prioritas rendah 4,2 detik, untuk 32 bit prosesor timer prioritas tinggi akan berputar kira-kira 4295 detik (1,2 jam), timer prioritas rendah kira-kira setiap 76 jam. Register timer dapat dibaca secara langsung, atau penjadwal dapat diinstruksi untuk mengantri sebuah proses untuk eksekusi setelah timer telah mencapai harga tertentu.

Antrian Timer diimplementasikan lagi sebagai link list dengan *workspace* dari proses menunggu, tetapi masukan diurut menurut waktu permintaan. Pointer depan pada antrian timer disimpan dalam lokasi-lokasi cadangan.

Gambaran lain yang penting tentang scheduler adalah kemampuannya untuk memilih salah satu dari sebuah group dari proses-proses untuk eksekusi, tergantung pada kemunculan event. Event-event itu mungkin lengkapnya dari sebuah masukan data, akhir periode timer atau interrupt eksternal.

3.3.2 Komunikasi Antar Proses

Untuk mendukung kerja paralelnya, transputer telah didesain dalam menggunakan *message-passing* komputer paralel. Transputer juga sangat didukung untuk *inter-process* komunikasi. Ini telah didesain dimana hampir tidak ada perbedaan dalam memprogram apakah itu komunikasi antara dua transputer yang sama atau berbeda.

Data ditransfer antara dua proses pada transputer yang sama dengan cara

menduplikat data antara ruang memori mereka. Sikronisasi dari transfer ini dengan melalui sebuah word kontrol kanal, yang merupakan sebuah word dimana saja dalam memori. Komunikasi hanya mengambil tempat ketika kedua proses masukan dan keluaran siap, waktu tunggu proses untuk komunikasi tidak mengambil waktu prosesor. Oleh karena itu kedua transputer disinkronisasikan dengan saling mengirimkan pesan antara mereka. Dua proses yang berhubungan melalui channel harus melibatkan lebar data yang sama, agar tidak terjadi suatu keadaan *deadlock*.

Komunikasi antara dua proses pada transputer yang berbeda menggunakan mekanisme yang sama, disinkronisasi oleh word kontrol kanal khusus yang terletak pada lokasi cadangan pada memori rendah. Instruksi yang sama digunakan untuk mengatur transfer, dan mesin link dalam setiap transputer menggunakan *Direct Memory Access (DMA)* untuk transfer antara link dan memori, tanpa campur tangan dari prosesor.

Setiap transputer mempunyai 2 atau 4 eksternal link. Komunikasi link secara *fullduplex* yang dapat menukar data dengan transputer lainnya dengan kecepatan 5, 10, 20 juta bit per detik, kecepatan dipilih melalui pin seleksi pada transputer. Kecepatan data dua arah dapat dicapai melebihi link tergantung pada tipe dari prosesor penukar data, dan kecepatan link, tetapi dapat ditingkatkan kira-kira 2,4 juta byte per detik untuk T805 transputer yang mempunyai link dengan kecepatan 20 juta bit per detik.

Data link ditransfer seperti sebuah serial urutan byte, setiap byte ditandai oleh transputer penerima. Tidak ada upaya yang diusahakan untuk mendeteksi link

pada transputer dengan anggapan medium komunikasi bebas kesalahan, atau komunikasi tingkat tinggi akan mendeteksi dan memperbaiki kesalahan.

Instruksi-instruksi channel adalah sebagai berikut:

ChanOut(), instruksi pendefinisian channel untuk pengiriman keluar proses dengan tipe data bebas.

ChanOutChar(), sama dengan *ChanOut()* dengan tipe data karakter.

ChanOutInt(), sama dengan *ChanOut()* dengan tipe data Integer.

ChanIn(), instruksi pendefinisian channel untuk penerimaan ke dalam proses dengan tipe data bebas.

ChanInChar(), sama dengan *ChanIn()* dengan tipe data karakter.

ChanInInt(), sama dengan *ChanIn()* dengan tipe data integer.

Jika diinginkan komunikasi yang lebih reliable dimana proses tidak akan terganggu dan keadaan *deadlock*, dapat dipakai komunikasi yang akan berhenti jika tidak memenuhi suatu kondisi yang diinginkan. Dilaksanakan dengan instruksi sebagai berikut:

ChanOutTimeFail(), instruksi akan melaksanakan pengiriman keluar proses sampai pada spesifikasi waktu yang ditentukan apabila pengiriman gagal maka pengiriman dihentikan dan channel direset.

ChanOutChanFail(), instruksi akan melaksanakan pengiriman keluar proses sampai suatu channel lain yang telah ditentukan (*failchan*) siap untuk menerima input, jika gagal pengiriman dihentikan dan channel direset.

ChanInTimeFail(), instruksi akan menerima pesan ke dalam proses sampai

spesifikasi waktu tertentu yang ditentukan, jika gagal penerimaan dihentikan dan channel direset.

ChanInChanFail(), instruksi akan menerima pesan ke dalam proses sampai suatu channel lain yang telah ditentukan (*failchan*) siap menerima input, jika gagal penerimaan dihentikan dan channel direset.

Pengalokasian dan inisialisasi channel dilaksanakan dengan instruksi *ChanAlloc()* dan *ChanReset()*. Khusus untuk 4 Link transputer hardware dapat berfungsi sebagai *channel bidirectional pointer* dengan spesifikasi alamat hardware yang berada pada library, yaitu:

```
#define LINK0OUT ((channel *) 0x80000000)
#define LINK1OUT ((channel *) 0x80000004)
#define LINK2OUT ((channel *) 0x80000008)
#define LINK3OUT ((channel *) 0x8000000C)
#define LINK0IN  ((channel *) 0x80000010)
#define LINK1IN  ((channel *) 0x80000014)
#define LINK2IN  ((channel *) 0x80000018)
#define LINK3IN  ((channel *) 0x8000001C)
```

3.3.3 Interrupt

Transputer mempunyai sebuah sumber dari interrupt eksternal. masukan **EventReq**. Pemrograman interface diimplementasikan pada masukan ini oleh karena itu kelihatan seperti word kontrol kanal lainnya, dilokasikan dalam memori rendah. Sebuah proses diinstruksikan untuk menunggu kanal event yang akan



dijadwal ulang sampai masukan **EventReq** menjadi '1', ketika dia akan di jadwal ulang. Ketika ini muncul keluaran handshake event **EventAck** dijadikan satu oleh prosesor.

Untuk tujuan memberi respon yang cepat pada interrupt proses menunggu untuk masukan event harus prioritas tinggi, dan merupakan proses prioritas tinggi satu-satunya yang jalan. Jika kondisi-kondisi ini dipertemukan maka waktu respon interrupt terburuk kira-kira 50 putaran prosesor. Respon interrupt cepat ini didapatkan dengan membuat instruksi-instruksi panjang yang dapat diinterrupt.

3.3.4 Memori

Anggota keluarga transputer mempunyai 2 atau 4 Kbytes internal RAM, lokasi terbawah dari ruang memori. beberapa disiapkan untuk fungsi kodemikro prosesor. Ini adalah jumlah memori yang tak sebanding dengan sebagian besar aplikasi-aplikasi, dan oleh karena itu transputer mempunyai eksternal memori. Membaca dan menulis pada internal memori lebih cepat kira-kira 1 putaran clock prosesor. Akses eksternal memori lebih lambat kira-kira 2 putaran prosesor dan 4 atau 5 dalam sistem dinamik memori.

Alamat memori adalah bertanda, maka memori transputer dimulai pada harga negatif terendah, **MinInt**, dimana #8000 untuk 16 bit transputer dan #80000000 pada 32 bit transputer. Ia akan berjalan dari nol harga positif terbesar, **MaxInt**, #7FFF atau #7FFFFFFF. Beberapa lokasi memori dicadangkan untuk fungsi-fungsi prosesor seperti terlihat pada tabel dibawah:

Tabel 3.2 ¹³
Lokasi Memori

Alamat Word	Name	Use
MintInt + 28	MemStart	T805,T800,T425,T225
...		dicadangkan untuk
...		instruksi lanjutan
MinInt + 18	MemStart	T414,T212,M212
MinInt + 17	EregIntSaveLoc	
MinInt + 16	STATUSIntSaveLoc	
MinInt + 15	CregIntSaveLoc	
MinInt + 14	BregIntSaveloc	daerah penyimpanan
MinInt + 13	AregIntSaveloc	register
MinInt + 12	IpPtrIntSaveloc	
MinInt + 11	WdescIntSaveLoc	
MinInt + 10	TptrLoc1	timer prioritas rendah
MinInt + 9	TPtrLoc0	timer prioritas tinggi
MinInt + 8	Event	kanal event
MinInt + 7	Link3Input	
MinInt + 6	Link2Input	
MinInt + 5	Link1Input	
MinInt + 4	Link0Input	kontrol word
MinInt + 3	Link3Output	link
MinInt + 2	Link2Output	
MinInt + 1	Link1Output	
MinInt	Link0Output	

Pada tabel ini alamat ditunjukkan sebagai offset word dari **MinInt**, Alamat byte sebenarnya tergantung pada panjang word dari transputer. Delapan word terakhir berisi word kontrol kanal untuk empat link eksternal, dengan word kontrol event pada lokasi yang kesembilan. Dua lokasi berikutnya berisi pointer depan untuk antrian proses timer prioritas tinggi dan prioritas rendah. Diatasnya lagi ada tujuh word yang menyimpan keadaan prosesor ketika proses prioritas rendah diinterrupt oleh proses prioritas tinggi.

¹³ Graham Ian, Op Cit, hal.15.

3.3.5 Pelayanan Sistem

Interface pelayanan sistem termasuk sinyal yang diperlukan untuk mereset dan mengeboot transputer, mengeset kecepatan dari prosesor dan link, mensinyal error dan merespon event eksternal. Transputer dapat direset dalam dua mode, tergantung harga dari sinyal **Analyse** ketika reset dinyatakan. Jika reset dengan **Analyse** tidak aktif transputer akan boot dari memori jika sinyal **BootFromRom** dihubungkan tinggi.

Jika **BootFromRom** adalah '0' akan memperhatikan link-linknya, dan dapat menerima sebuah program bootstrap atau perintah untuk menulis (*poke*) atau membaca (*peek*) lokasi memori melalui link. Masukan **Analyse** digunakan untuk mempermudah proses debug.

Jika **Analyse** menjadi '1' sementara transputer berjalan maka dia akan memberhentikan secepat mungkin, dan link akan menjadi tidak aktif. Kemudian terjadi **Reset**. Ketika **Reset** menjadi '0' lagi transputer akan tetap dalam keadaan reset, tetapi register akan tetap berisi informasi keadaan mesin sebelum dihentikan oleh **Analyse**.

3.4 Karakteristik Transputer IMS T805

Teknologi transputer sendiri lahir sekitar awal tahun 1980an. Transputer IMS T805 adalah sebuah komputer mikro CMOS dengan arsitektur 32 bit, dengan sebuah unit floating point 64 bit yang sesuai dengan standard IEEE 754. Transputer mempunyai 4 Kbytes RAM statis internal (*on-chip*) yang mempunyai kecepatan tinggi, sebuah interface memory eksternal yang dapat dikonfigurasi

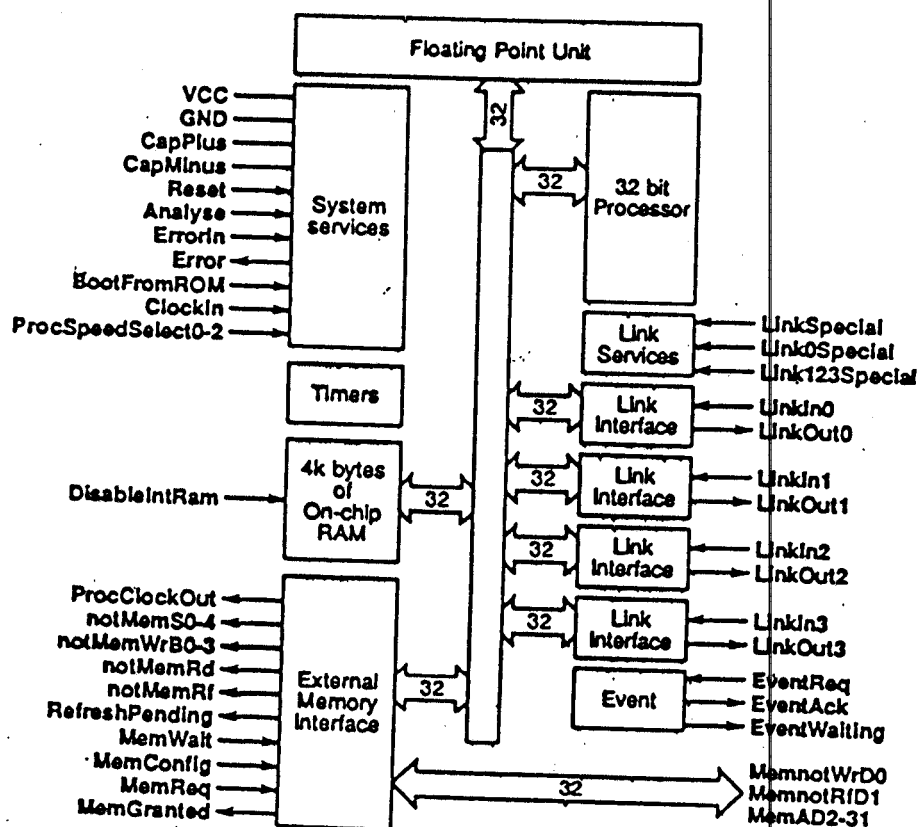
sampai 4 Gbytes dan 4 link komunikasi serial standard INMOS yang dapat diatur kecepatannya pada 5/10/20 Mbits/sec. Kecepatan tiap link komunikasi serial dapat mencapai 2,4 Mbytes/sec *bi-directional*.

Perintah-perintah mnemonik lebih efisien jika diimplementasikan pada bahasa tingkat tinggi, perintah-perintah ini sangat mendukung pemrograman *concurrent* pada bahasa pemrograman OCCAM baik ketika menggunakan sebuah transputer atau dalam sebuah jaringan transputer. Pemanggilan prosedur, proses *switching* dan interupsi pada umumnya membutuhkan waktu tidak lebih dari orde mikrodetik.

Kecepatan prosesor dari transputer dapat dipilih melalui pin seleksi mulai dari 17,5 MHz sampai kepada maksimum yang dibolehkan untuk transputer. Transputer yang bekerja pada 30 MHz melaksanakan sebuah instruksi dengan kecepatan puncak 30 MIPS (Million Instructions Per Second atau juta instruksi per detik) dan terendah 15 MIPS.

Transputer IMS T805 menghasilkan unjuk kerja yang tinggi untuk operasi aritmatik dan floating point. Unit 64 bit floating point menghasilkan operasi presisi tunggal atau ganda pada standar ANSI-IEEE 754-1985 untuk aritmatik floating point. Operasi floating point dapat dilaksanakan secara *concurrent* dengan prosesor, dengan kecepatan 2,2 Mflops pada kecepatan prosesor 20 MHz dan 3,3 Mflops pada 30 MHz.

3.4.1 Prosesor



Gambar 3.3 ¹⁴
Blok Diagram Transputer IMS T805

Prosesor 32 bit yang berisi instruksi proses logika, pointer instruksi dan work, dan sebuah register operand. Prosesor secara cepat dapat mengakses 4 Kbyte memori on-chip atau disebut juga internal memori yang dapat digunakan untuk menyimpan data atau program. Bila jumlah memori yang dibutuhkan besar maka prosesor dapat mengakses sampai 4 Gbyte memori melalui *External Memory Interface EMI*.

¹⁴ _____, *Transputer Technical Specifications*, Computer System Architects (CSA), USA, 1990, hal.3.

3.4.2 Floating Point Unit (FPU)

64 bit Floating Point Unit (FPU) menyediakan aritmatik floating point (FP) 32 bit dan 64 bit standard ANSI-IEEE 754-1985. Ini memungkinkan operasi aritmatik FP secara *concurrent* dengan prosesor utama (CPU), dengan kecepatan 3,3 Mflops pada 30 MHz. Semua komunikasi data antara memori dan FPU dibawah kendali CPU.

FPU terdiri dari sebuah kodemikro mesin komputer dengan 3 stack evaluasi FP untuk manipulasi dari bilangan-bilangan FP Stack register sebagai berikut FA, FB dan FC, masing-masing dapat menerima 32 bit atau 64 bit data. Sebuah *flag* set ketika sebuah harga FP dimasukkan, Kelakuan stack seperti pada stack CPU.

Seperti stack CPU, stack FPU tidak disimpan ketika penjadwalan muncul. FPU dapat digunakan pada proses prioritas rendah atau tinggi. Ketika proses prioritas tinggi menginterup prioritas rendah keadaan FPU disimpan dalam FPU. CPU segera melayani interup ketika sedang menyelesaikan operasi yang sedang berlangsung. Proses prioritas tinggi tidak akan mulai, sebelum FPU telah menyelesaikan operasi yang sedang berlangsung.

Penunjuk dalam sebuah urutan instruksi dimana data perlu ditransfer ke atau dari FPU disebut sinkronisasi penunjuk. Pada sebuah penunjuk sinkronisasi unit proses yang lebih dahulu siap akan menunggu sampai yang lain siap. Data yang ditransfer kemudian muncul dan kedua prosesor akan berjalan secara *concurrent* lagi.

Untuk memaksimalkan proses *concurrent*, data FP dan alamat tujuan

dapat dihitung oleh CPU sementara FPU melakukan operasi pada data sebelumnya. Untuk mengoptimasi alat dilakukan dengan cara meminimumkan waktu idle CPU dan FPU.

FPU telah didesain untuk operasi bilangan floating point 32 bit dan 64 bit, dan mengembalikan hasil sesuai standard ANSI-IEEE 754-1985. Bilangan yang dinormalisasi kembali sepenuhnya didukung oleh perangkat keras.

FPU mempunyai flags kesalahan sendiri yaitu **FP_Error**. Flag ini mencerminkan keadaan evaluasi dari FPU dan akan diset jika terjadi kesalahan. **FP_Error** juga diset jika sebuah input floating point adalah tak terbatas atau bukan sebuah bilangan. **FP_Error** dapat diset, dites, dibersihkan tanpa mempengaruhi flag **Error** utama, tetapi flag **Error** dapat diset ketika dibutuhkan.

3.4.3 Memori

Transputer IMS T805 mempunyai 4 Kbytes memori statik internal kecepatan tinggi untuk meletakkan data. Akses setiap memori internal memakan satu putaran prosesor *ProcClockOut*. Transputer juga dapat mengakses 4 Gbytes ruang eksternal memori. Internal(*on-chip*) dan eksternal(*off-chip*) memori adalah bagian yang sama dari ruang alamat linier. Internal ram dapat tidak diaktifkan dengan menjaga *DisableIntRam* menjadi '1'. Semua alamat-alamat internal akan dipetakan kepada memori eksternal. Pin ini tidak perlu diubah setelah *Reset* menjadi '0'.

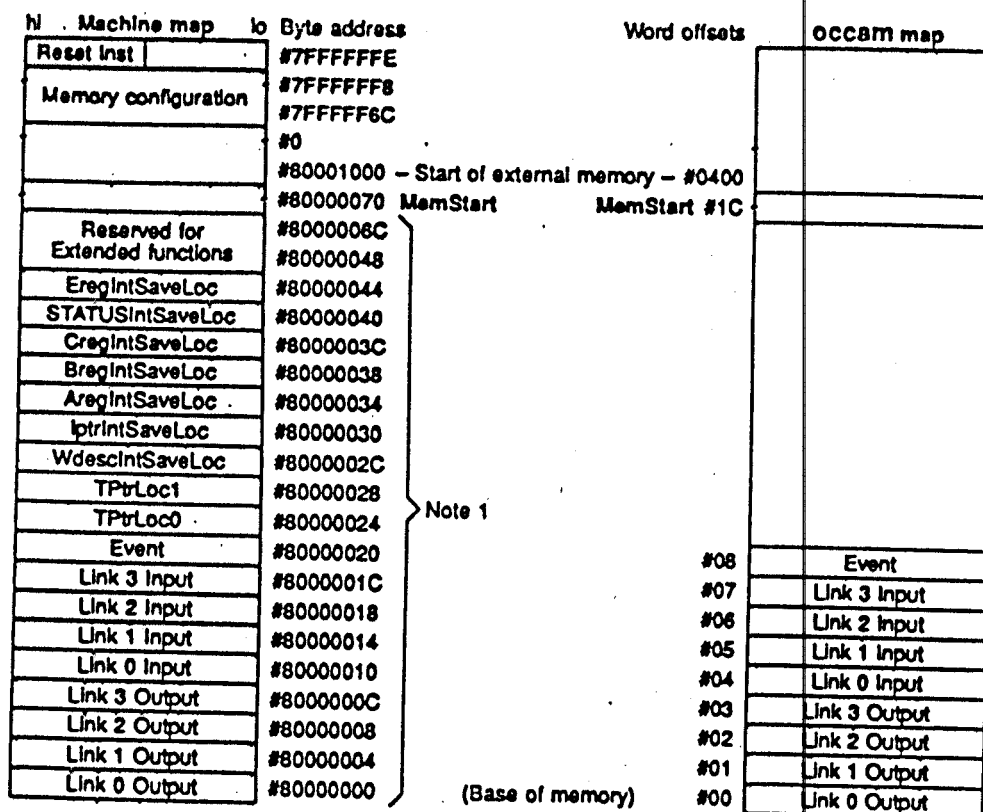
Memory IMS T805 adalah pengalamatan byte (*Byte Addressed*), dimana empat byte membentuk satu word. Byte akhir dari sebuah word adalah

pengalamatan byte terendah.

Bit-bit dalam sebuah byte diberi nomor dari 0 sampai 7. dengan bit 0 adalah bit terendah. Byte-byte diberi nomor dari 0, dengan byte 0 adalah byte akhir. Umumnya dimanapun sebuah harga diperlakukan sebagai sejumlah array data, dimana komponen-komponen data diberi nomor untuk kenaikan indeksnya, dengan komponen terendah diberi nomor 0. Dimana harga-harga disimpan dalam memori, dan indeks terendah dari komponen disimpan pada alamat terendah (biasanya negatif).

Memori internal dimulai dari alamat negatif #80000000 sampai #80000FFF (untuk tipe T805). Memori yang dapat digunakan mulai pada #80000070, lokasi ini dinamakan *MemStart*. Instruksi *ldmemstartval* digunakan untuk mendapatkan harga dari *MemStart*.

Sebuah proses dalam transputer melibatkan sebuah *workspace descriptor* (WPtr) dan sebuah pointer instruksi (IPtr). WPtr adalah sebuah pointer alamat dalam word pada *workspace* di memori. IPtr menunjuk pada instruksi berikutnya yang akan dieksekusi untuk proses yang sedang dijalankan. *Context switch* dibentuk oleh instruksi breakpoint dengan jalan menukar WPtr dan IPtr dari proses yang sedang berlangsung dengan WPtr dan IPtr yang berada di atas *MemStart*. Dua *context* diletakkan di atas *MemStart*, satu untuk prioritas tinggi dan satu untuk prioritas rendah, ini memungkinkan pemrosesan pada kedua level untuk mendapatkan breakpoint.



Gambar 3.4 ¹⁵
Peta Memori Transputer IMS T805

Perhatikan pada bootstrapping dari sebuah link, *context-context* itu ditumpuki oleh kode yang di-load. Jika ini tidak diinginkan, harga-harga dapat diambil dari memori sebelum *bootstrapping* dari sebuah link.

Area yang dicadangkan dari memori internal dibawah MemStart digunakan untuk mengimplementasikan link dan kanal-kanal event.

Dua word dari memori dicadangkan untuk penggunaan timer, TPtrLoc0 untuk proses-proses prioritas tinggi dan TPtrLoc1 untuk proses-proses prioritas

¹⁵ Ibid hal.33.

rendah. Mereka juga mengindikasikan prioritas timer relevan tidak digunakan atau menunjuk pada proses pertama pada antrian timer pada level prioritas.

Harga dari register-register prosesor tertentu untuk proses prioritas rendah disimpan dalam lokasi-lokasi cadangan IntSaveLoc ketika sebuah prioritas tinggi memproses pre-empt yang merupakan prioritas rendah. Lokasi yang lain disiapkan untuk perluasan-perluasan seperti block moves dan operasi floating point.

Eksternal memori mulai pada #80001000 dan meluas dari #00000000 sampai #7FFFFFFF. Konfigurasi data memori dan ROM bootstrapping code harus dalam alamat positif, mulai pada #7FFFFFF6C dan #7FFFFFFE. Ruang alamat dibawah ini adalah digunakan secara konvensional untuk kode dasar ROM.

3.4.4 *Link*

Empat serial link dua arah yang identik memberikan komunikasi yang sinkron antara prosesor dengan dunia luar. Setiap link berisi sebuah kanal masukan dan kanal keluaran. Sebuah link antara dua transputer diimplementasikan dengan menghubungkan sebuah interface link pada satu transputer ke interface link transputer lainnya. Setiap byte data yang dikirim melalui sebuah link dikenali oleh masukan dari link yang sama, oleh karena itu setiap signal membawa data dan informasi kontrol.

Keadaan diam dari sebuah link output adalah '0'. Setiap data byte ditransmisikan sebagai *start bit* berlogika '1' diikuti oleh sebuah bit diikuti oleh delapan bit data dan diikuti oleh *stop bit* berlogika '0'. Bit terendah dari data ditransmisikan dahulu. Setelah pengiriman sebuah byte data, pengirim menunggu

sinyal pengenalan, yang terdiri dari sebuah *start bit* berlogika '1' diikuti oleh sebuah bit nol. Sinyal pengenalan menandakan sebuah proses sudah dapat menerima data byte dengan benar dan link penerima sudah dapat menerima byte yang lain. Link pengirim menjadwalkan ulang proses pengiriman hanya setelah sinyal pengenalan untuk byte terakhir dari pesan telah diterima.

Link transputer IMS T805 mendukung komunikasi dengan kecepatan standard 10 Mbits/dtk. Link transputer juga dapat digunakan pada 5 atau 20 Mbits/dtk untuk kecepatan prosesor transputer 17 MHz, 20 MHz dan 25 MHz dan 20 Mbits/dtk transputer yang lebih cepat. Link tidak disinkronisasikan dengan *ClockIn* atau *ProcClockOut* dan mengabaikan phase-phasesnya. Maka link yang berasal dari sistem clock yang terpisah dapat berkomunikasi, dengan syarat clock secara nominal identik dan masih dalam spesifikasi.

3.5 Aplikasi Transputer IMS T805

- Aplikasi ilmu pengetahuan dan matematika
- Sistem *multi processor* kecepatan tinggi
- Pengolah grafik unjuk kerja tinggi
- Super komputer
- *Workstation* dan *workstation cluster*
- *Digital Signal Processing (DSP)*
- *Distributed database*
- Pemercepat prosesor / *Accelerator processors*
- Simulasi sistem

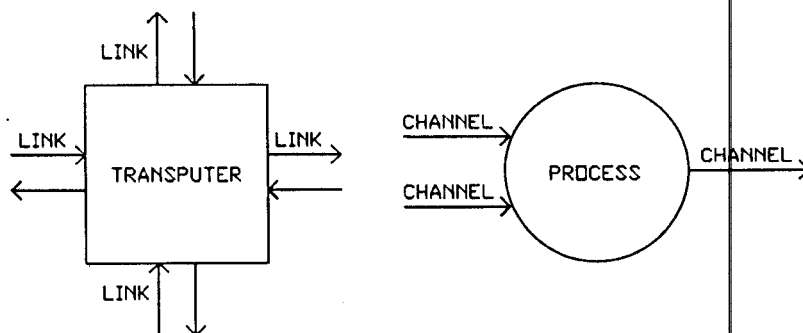
- Telekomunikasi
- Sistem toleransi kesalahan
- *Robotics*
- *Pattern recognition*
- Pengolah citra / *Image processing*
- Kecerdasan buatan / *Artificial Intelligence (AI)*

3.6 Konfigurasi

Konfigurasi merupakan rangkaian atau susunan dimana proses ditempatkan pada prosesor (transputer) dan kanal.

3.6.1 Prosesor dan Proses

Sebuah link merupakan dua kanal (dua alamat atau dua penghubung), masing-masing satu arah. IMS T805 mempunyai empat link sehingga terdapat delapan kanal, empat masuk dan empat keluar. Sebuah link tidak dapat dipisah, misal kanal input dari salah satu prosesor dan kanal output ke lainnya, karena jika salah satu alamat atau penghubung tidak digunakan, transputer masih menggunakannya untuk *handshaking* dan sinkronisasi.



Gambar 3.5 ¹⁶
Prosesor dan Proses

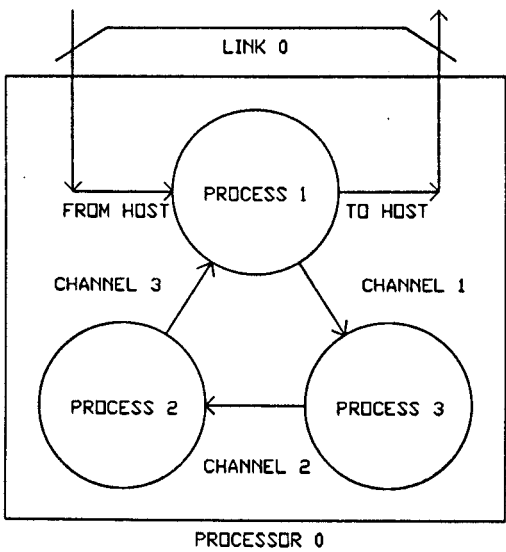
3.6.2 Konfigurasi Transputer ¹⁷

Untuk semua contoh disini, diasumsikan Link 0 terhubung ke *host* (PC), dengan maksud diperlukan pada proses *booting* transputer. Dalam contoh dimisalkan terdapat tiga proses, maka ketiga proses tersebut dapat dioperasikan pada satu, dua atau tiga prosesor.

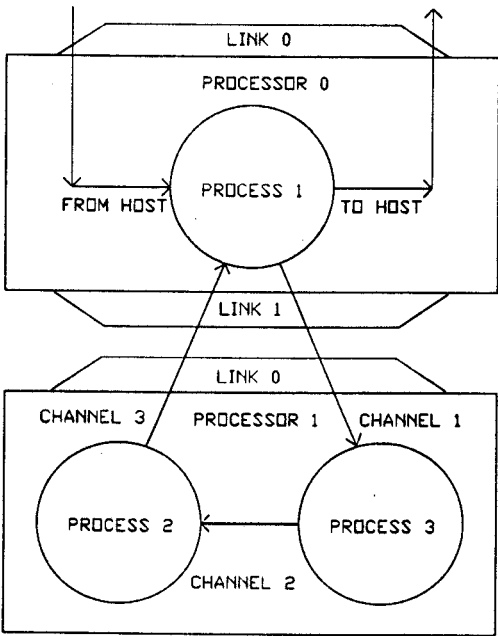
Namun dalam tugas akhir ini hanya digunakan satu transputer. Disini diberi contoh dengan maksud untuk pengembangan lebih lanjut.

¹⁶ _____, *Occam and the Transputer a Workbook*, Computer System Architects (CSA), USA, 1990, hal.83.

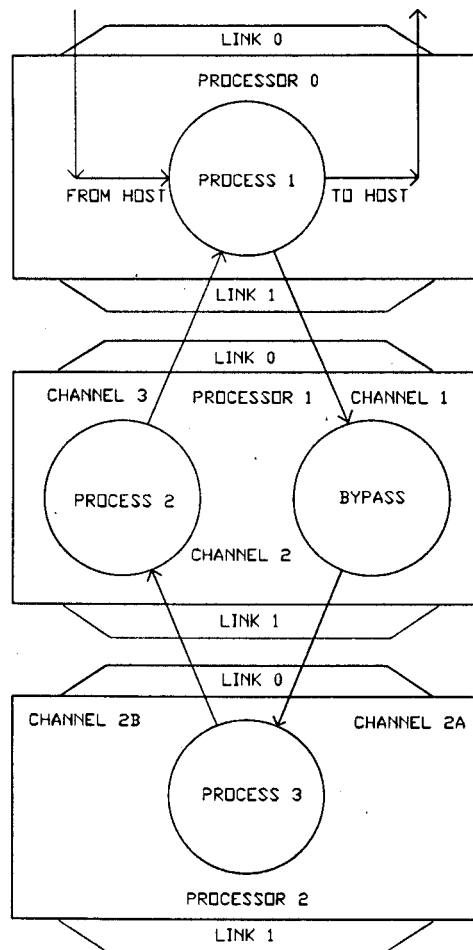
¹⁷ Loc Cit.



Ganbar 3.6
Konfigurasi Satu Transputer



Gambar 3.7
Konfigurasi Dua Transputer



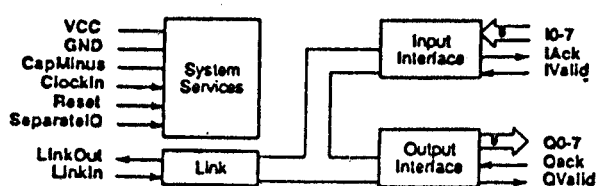
Gambar 3.8
Konfigurasi Jaringan Transputer

3.7 Serial Link Interface

Salah satu keunggulan dari transputer adalah tidak diperlukannya banyak komponen pendukung untuk dapat berfungsi dalam komunikasi dan memori interface. INMOS telah membuat 2 model dari komponen pendukung, adaptor link digunakan untuk menghubungkan transputer dengan sistem orientasi-bus konvensional dan *switch link*, yang dapat memungkinkan link konfigurasi elektronik dengan transputer.

3.7.1 Adaptor Link

Untuk menghubungkan sebuah link transputer pada komponen paralel, seperti bus konvensional mikroprosessor, maka sebuah adaptor link dibutuhkan. Ada dua tipe adaptor link yaitu INMOS C011 dan C012, tetapi dalam tugas akhir ini yang dipakai adalah tipe INMOS C011 dengan mode operasi 1. Blok diagram dari komponen-komponen tersebut dapat dilihat pada gambar dibawah ini:



Gambar 3.9 ¹⁸
Block Diagram C011 Mode 1

IMS C011 Mode 1 :

Mempunyai I/O 8 bit data paralel terpisah. Dalam mode ini IMS C011 mengkonversi antara sebuah link dan dua interface 8 bit dengan *handshake*, satu untuk input dan satu output.

3.7.2 C011 Mode 1

Dalam mode ini C011 dikonfigurasi sebagai peripheral interface dengan port input terpisah dari port output. Pin *IValid* dan *Iack* adalah pin handshake untuk memasukkan data dari peripheral ke transputer.

¹⁸ _____, *Transputer Serial Link Interfacing Kit IMS C011 Link Adaptor Engineering Data*, Computer system Architects (CSA), USA, 1990, hal.390.

Ketika data pada pin I0-I7 valid, *IValid* diberi sinyal '1' oleh komponen peripheral. Link kemudian meneruskan atau mengirim data yang ada pada pin-pin I0-I7, dan ketika transfer data selesai maka *IAck* akan berlogika '1'. Ketika komponen peripheral mengirim sinyal '0' pada *IValid*, *IAck* akan menjadi '0' oleh C011. Sebuah byte data yang diterima oleh C011 akan diteruskan pada port output Q0-Q7, dan *QValid* akan berlogika '1' oleh C011.

Setelah data dibaca oleh peripheral maka peripheral harus mengirim sinyal '1' pada pin *QAck*. C011 kemudian akan mengirim sebuah paket *acknowledgment* pada link untuk memberi signal bahwa transfer data sudah selesai. C011 akan mengeset *QValid* '0', dan peripheral dapat mengirim sinyal '0' pada *QAck* lagi. Fungsi-fungsi pin pada IMS C011 mode 1.

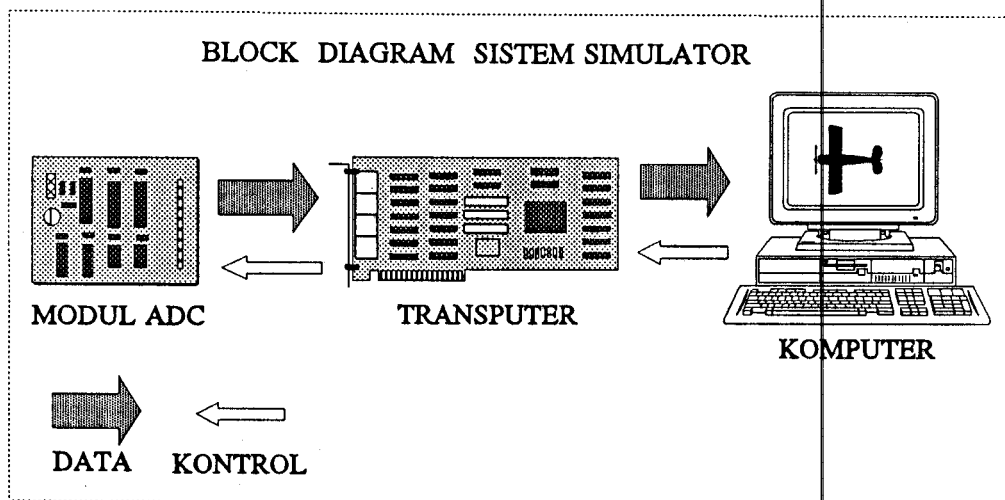
Pin	I/O	Fungsi
LinkIn	in	data dari link
LinkOut	out	data ke link
I0-I7	in	input data dari bus
IValid	in	indikasi data valid ditulis dari bus ke I0-I7
IAck	out	indikasi data input sukses ditransmisikan ke link
Q0-Q7	out	output data ke bus
QValid	out	indikasi data valid berada dalam register output
QAck	in	indikasi bahwa data output telah dibaca.

BAB IV

PERENCANAAN HARDWARE

4.1 Perencanaan Hardware Sistem

Blok diagram secara umum dari keseluruhan perencanaan hardware sistem seperti gambar 4.1. Terdapat dua arah anak panah yang menggambarkan aliran data dan kontrol. Kontrol berasal dari komputer ke transputer dan dari transputer ke modul ADC. Sedangkan data dari modul ADC ke transputer dan setelah diolah hasilnya dikirim ke komputer untuk ditampilkan.

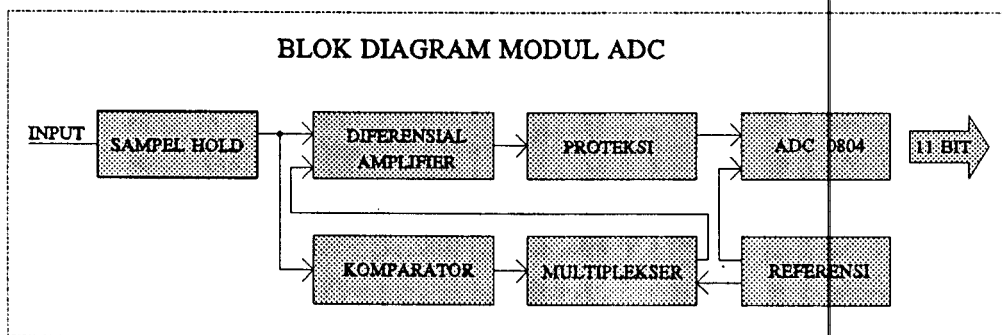


Gambar 4.1
Blok Diagram Sistem

Sedangkan blok diagram dari perencanaan modul ADC yang lebih terperinci terdapat pada gambar 4.2.

4.2 Perencanaan Pengolah Sinyal Analog

Pengolah sinyal analog akan mengolah secara analog tegangan input yang



Gambar 4.2
Blok Diagram Modul ADC

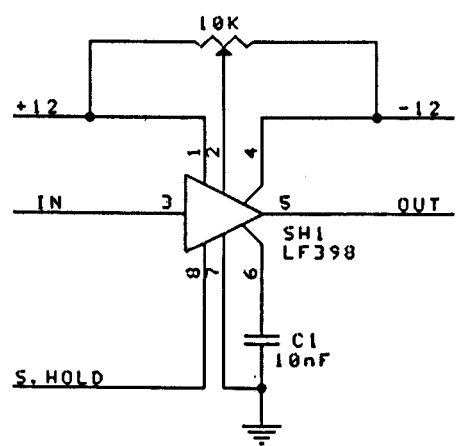
akan diproses sebelum diumpankan ke ADC. Pengolahan ini perlu supaya sinyal input yang masuk ke ADC memenuhi batas tegangan yang dianjurkan dan dengan demikian maka keamanan ADC dapat dijamin. Selain itu untuk memperkecil kesalahan hasil konversi karena adanya laju perubahan tegangan.

4.2.1 Penguat Sample-Hold

Rangkaian sample-hold seperti terlihat pada gambar 4.3 menggunakan IC LF 398N sebagai penguat sample-hold dimungkinkan menyampling sinyal analog dan nantinya dikonversikan ke bentuk digital dengan sedikit kesalahan.

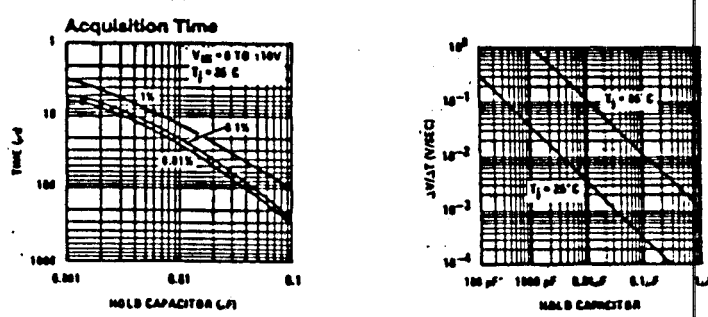
Rangkaian penguat sample-hold bekerja sebagai berikut. Bila tegangan input kontrol tinggi, rangkaian dalam mode sample. Dalam hal ini penguat berfungsi seperti rangkaian buffer atau voltage follower biasa. Waktu sample minimum ditentukan oleh waktu acquisition dari LF 398N.

Bila input tegangan rendah, rangkaian dalam mode hold dan akan mempertahankan tegangan yang telah ada pada kapasitor. Batas maksimum waktu hold ini ditentukan oleh perbandingan antara resolusi ADC (dalam Volt) terhadap



Gambar 4.3
Rangkaian Sample Hold

laju droop penurunan tegangan output sebagai fungsi waktu.



Gambar 4.4¹⁹
Waktu Acquisasi dan Laju Drop Tegangan Terhadap Capacitor Hold

Berdasarkan grafik waktu acquisition yang dikeluarkan pabrik pembuat LF-398N, kapasitor hold sebesar 10 nF memberi waktu acquisition sebesar 11 μ s. Waktu ini bersama-sama dengan waktu konversi ADC menentukan waktu sampling minimum yang diijinkan. Waktu sampling maksimum ditentukan oleh laju droop penguat sample-hold yaitu dengan kapasitor hold 10 nF diperoleh laju

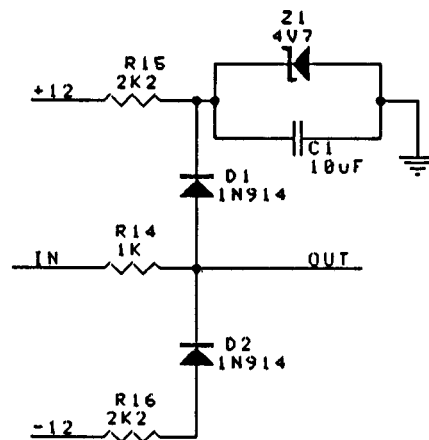
¹⁹ _____, *Linier Databook 2*, National Semiconductor Corporation, 1988, hal.5-7.

droop sebesar 3,5 mV/detik. Karena resolusi ADC yang direncanakan sebesar $512\text{mV}/256 = 2\text{ mV}$, maka waktu sampling maksimum adalah:

$$T = \frac{\text{resolusi ADC}}{\text{lajudroop}} = \frac{2\text{ mV}}{3.5\text{ mV/detik}} = 0,57\text{ detik}$$

4.2.2 Proteksi

Rangkaian proteksi ditempatkan pada bagian akhir pengolah sinyal analog karena berfungsi untuk melindungi rangkaian selanjutnya terhadap tegangan lebih. Rangkaian ADC yang direncanakan mempunyai daerah tegangan input 0 - +5 V sedangkan output pengolah sinyal analog dapat mempunyai tegangan di atas +5V.



Gambar 4.5
Rangkaian Proteksi

Rangkaian proteksi bekerja bila tegangan pada titik pertemuan D₁ dan D₂ melebihi 5,4 V, yaitu penjumlahan tegangan forward diode 1N914 sebesar 0,7 V dan tegangan kerja diode zener 4,7 V. Tahanan R₁₅ dan R₁₆ berfungsi memberikan bias arus bagi D₁ dan D₂ agar bekerja normal. Tahanan pembatas arus R₁₄ sebesar 1 KΩ akan mengamankan output tahap sebelumnya bila tegangan melebihi batas.

4.2.3 Auto Range

Untuk mendapatkan resolusi yang lebih tinggi digunakan pembagian range pada sinyal input. Dalam desain rangkaian sinyal input dibagi menjadi 8 range (3 bit), tiap-tiap range besarnya 512 mV.

4.2.3.1 Differential Amplifier

Differential amplifier disusun dari sebuah op-amp LF 356 dengan besar penguatan satu kali. Penguatan disini digunakan untuk menguatkan beda tegangan antara kedua input op-amp yaitu selisih antara non-inverting input dengan inverting input.

Tegangan input non-inverting berasal dari sinyal input yang akan dikonversikan, sedangkan tegangan input inverting berasal dari output auto range. Jadi tegangan yang akan dikuatkan adalah besar tegangan sinyal input dikurangi tegangan pengurang dari output multiplexer.

4.2.3.2 Komparator

Rangkaian ini pada intinya terdiri dari 7 komparator serta sebuah saklar analog (multiplexer), membentuk satu kesatuan rangkaian flash ADC 3 bit. Ada 7 komparator yang dibentuk oleh dua IC LM 324 dan kemudian outputnya diumpankan sebagai data bagi saklar multiplexer. Output komparator diatur sedemikian hingga berlevel sekitar +5 V atau 0 V (level TTL).

Multiplexer yang dipakai dari jenis IC CMOS 4051. Tegangan input yang telah disesuaikan levelnya dibandingkan oleh rangkaian komparator untuk

mendapatkan kode bit pemilih pada 4051 untuk memilih level tegangan pengurang yang sesuai. Kemudian output 4051 ini diumpankan ke input (-) differential amplifier.

4.2.3.3 Encoder Auto Range

Rangkaian encoder berguna untuk mengkonversi data output 7 komparator menjadi kode biner tiga bit dan menjadi bit kontrol bagi multiplexer untuk memilih salah satu kanal inputnya. IC yang digunakan adalah jenis TTL yaitu 74LS147.

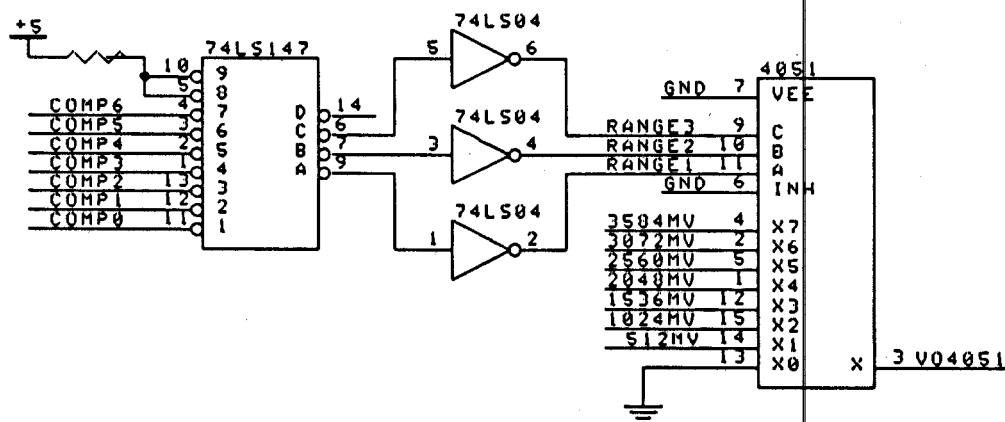
Untuk menyesuaikan logika agar dapat bekerja sesuai dengan perencanaan rangkaian, diperlukan 3 inverter 74LS04 pada output encoder. Tiga bit output encoder digunakan sebagai selector pada multiplexer.

4.2.3.4 Multiplexer

Multiplexer difungsikan untuk menyeleksi tegangan yang telah digunakan sebagai pembanding dalam dalam rangkaian komparator, nantinya tegangan hasil selektor ini dipakai untuk mengurangi sinyal input.

Jumlah tegangan yang diseleksi sebanyak 7, sesuai dengan jumlah komparator atau range yang digunakan. Diawali dengan tegangan 0 V (ground) kemudian naik tiap step sebesar 512 mV, dapat dilihat pada gambar 4.6

Jadi dalam rangkaian ini besar tegangan yang masuk ke input ADC tidak akan lebih dari 512 mV, sebab sinyal input dikurangi tegangan pengurang berkelipatan 512 mV. Dalam arti apabila tegangan input melebihi 512 mV akan



Gambar 4.6
Rangkaian Multiplexer

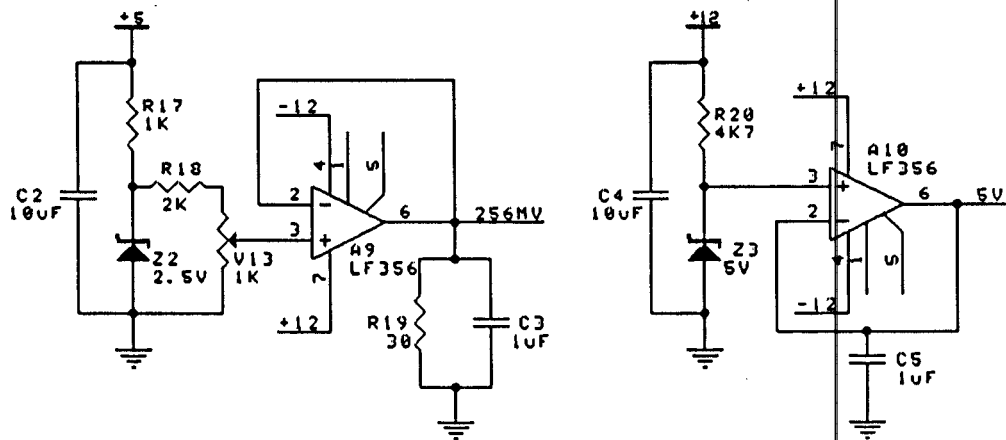
terjadi perpindahan range, dan tegangan pengurang akan diganti lebih besar.

4.2 Tegangan Referensi

Modul hardware yang direncanakan membutuhkan dua tegangan referensi masing-masing 5 V dan 256 mV. Tegangan 5 V diperlukan untuk referensi komparator pada autorange, sedangkan tegangan 256 mV diperlukan sebagai referensi ADC 0804 (pin Vref/2). Komponen utama pada tegangan referensi adalah zener diode referensi LM 336 5V dan LM 336 2,5V, keduanya mempunyai range arus revers dari 400 μ A sampai 10 mA.

Tegangan 5 V didapat dari zener LM 336 5V beserta rangkaian buffer A₁₀ op-amp LF 356. Zener diberi bias revers dari tegangan 12 V melalui resistor 4K7, sehingga arus bias reversnya adalah $(12-5)/4700=1,5$ mA. Besar arus ini masih memenuhi persyaratannya atau di dalam range yang diijinkan.

Sedangkan tegangan 256 mV didapat dari hasil pembagian tegangan oleh



Gambar 4.7
Rangkaian Tegangan Referensi

R_{18} dan V_{13} , yang diatur sedemikian sehingga output buffer A_9 mempunyai tegangan 256 mV. Zener LM 336 2.5V diberi bias revers dari 5 V melalui resistor 1K, sehingga arus yang lewat R_{17} adalah $(5-2,5)/1000 = 2,5$ mA. Sedangkan arus revers zener adalah $2,5 - (2,5/(2K+1K)) = 1,67$ mA. Besar arus ini masih memenuhi range yang diijinkan.

Untuk menjaga kesetabilan tegangan, sumber tegangan bias diberi kapasitor 10 μ F dan output buffer diberi kapasitor sebesar 1 μ F paralel resistor 30 Ω .

4.3 Rangkaian ADC

4.3.1 ADC 0804

Untuk mengkonversi sinyal analog menjadi data digital diperlukan unit konverter dan dalam perencanaan ini digunakan ADC dari jenis *Successive Aproximation* yaitu type ADC 0804. IC ini mempunyai satu channel input dan 8 bit output.

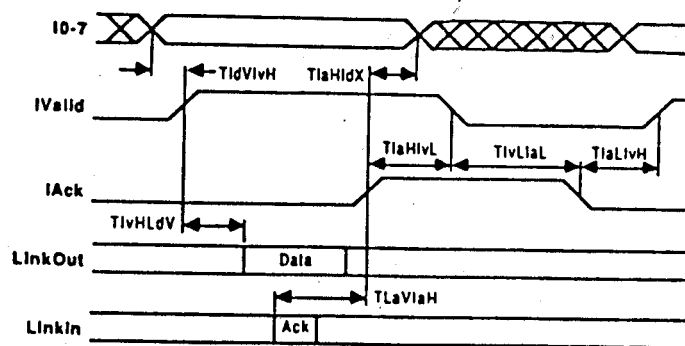
ADC ini difungsikan untuk input 0-512 mV, sehingga resolusinya

$$\frac{512 \text{ mV}}{2^8} = \frac{512}{256} = 2 \text{ mV}$$

Supaya dapat mengkonversi sampai tegangan 512 mV maka pada pin Vref/2 diberi tegangan referensi sebesar $V_{in \text{ max}} / 2 = 256 \text{ mV}$.

4.3.2 Selector Range dan Data

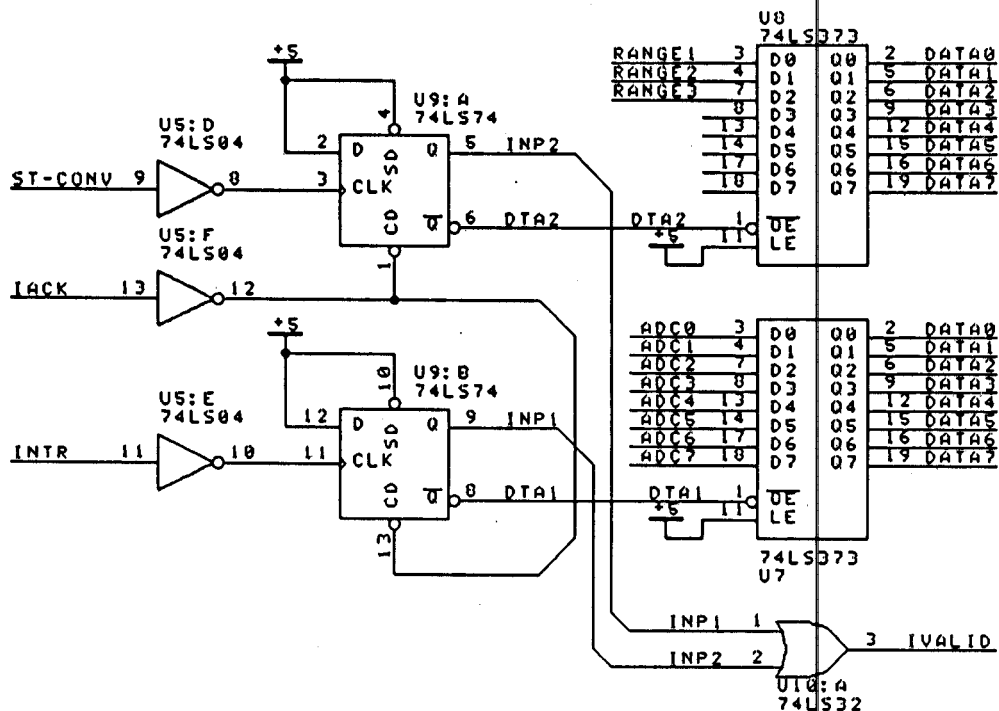
Guna menangani pengambilan antara range dan data diperlukan chip select antara data latch untuk range dan data latch untuk hasil konversi ADC. Supaya memenuhi time diagram pada gambar 4.8, rangkaian disusun dari dua D flip-flop dan tiga inverter.



Gambar 4.8²⁰
Time Diagram IMS C011 Mode 1

Dari rangkaian gambar 4.9 apabila ST_CONV (start konversi) berubah dari logika '1' menjadi logika '0' maka ADC 0804 mulai mengkonversi sinyal input,

²⁰ _____, *Transputer Serial Link Interfacing Kit IMS C011 Link Adaptor Engineering Data*, Ibid, hal.389



Gambar 4.9
Rangkaian Selector Range dan Data

dan saat itu juga D flip-flop U₉:A mendapat clock sehingga Q (INP2) berlogika '1' dan invers Q (DTA2) berlogika '0'. Status ini memberikan arti bahwa data latch U₈ aktif atau siap dan IValid berlogika '1' atau data sudah Valid untuk diambil. Karena yang aktif data latch U₈ maka yang diambil adalah data range.

Setelah kira-kira 100 μ S ADC telah selesai mengkonversi dan memberi logika INTR dari '1' ke '0'. Perubahan ini memberi input clock pada D₉:B. Perubahan logika INTR ini menjadikan data latch U₇ aktif dan IValid berlogika '1'. Kondisi ini berarti data sudah valid dan siap diambil, data ini berupa hasil konversi sinyal yang masuk ke ADC. Dengan selesainya dua proses di atas yaitu pengambilan range dan data lengkaplah satu siklus pengambilan data.

4.4 Serial Interface

Untuk menghubungkan modul ADC ke prosesor diperlukan rangkaian interface. Desain rangkaian yang dibuat menggunakan IC C011-M1 serial to parallel interface. IMS C011 beroperasi pada mode 1 dengan clock 5.0 MHz dan menghubungkan pin *SeparateIQ* ke Vcc. Ada empat bit data untuk mengatur sinkronisasi lalu lintas data yang akan masuk maupun keluar dari serial link interface yaitu QAck, QValid, IAck dan IValid.

Seperti gambar rangkaian pada lampiran QAck dihubungkan langsung dengan QValid, hal ini menjadikan status output selalu valid dan setiap mengirim data tanpa proses tunggu (*free running*). Sedangkan pengambilan data masih menunggu sampai IValid berlogika '1', apabila masih berlogika '0' berarti data belum siap.

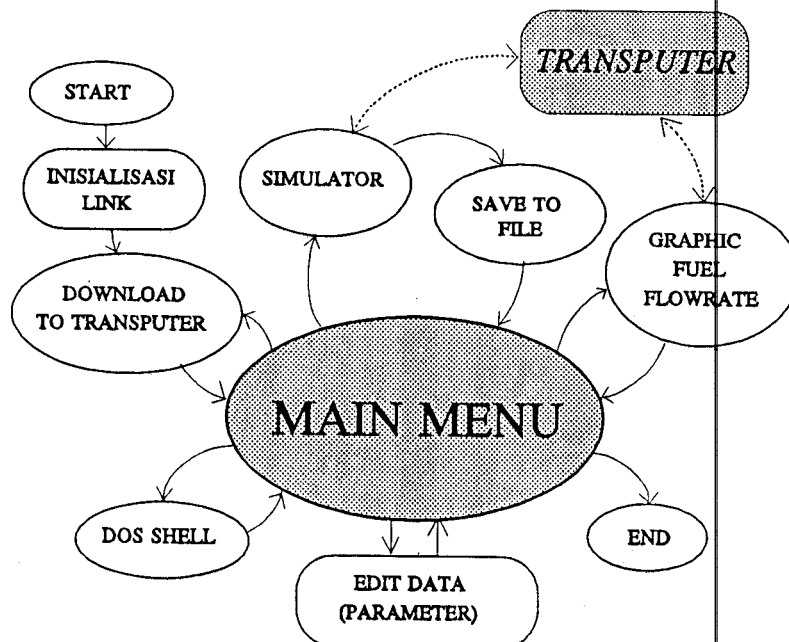
Sebagai jalur komunikasi antara serial link interface dengan rangkaian ADC digunakan kabel paralel 40 baris, dan untuk menghindari *crosstalk* masing-masing jalur data disekat oleh jalur ground, disusun selang-seling antara data dan ground.

BAB V

PERENCANAAN SOFTWARE

Setelah perencanaan hardware dilakukan maka selanjutnya perencanaan software memegang peranan penting, sebab software disini merupakan inti dari sistem dan bukan hanya membantu hardware.

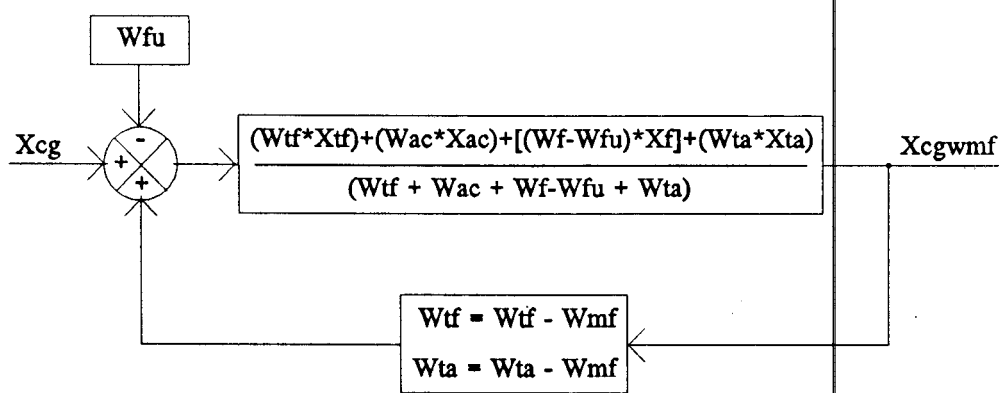
Software yang direncanakan dibagi menjadi dua bagian. Satu beroperasi di PC AT 386DX dan satu lagi beroperasi di Transputer T805. Software yang beroperasi di PC disusun dengan bantuan compiler Borland C++ versi 3.1, sedangkan software yang beroperasi di Transputer disusun dengan bantuan compiler Logical System C89.1. Adapun flow diagram sistem yang direncanakan seperti gambar berikut.



Gambar 5.1
Flow Diagram Software Sistem

5.1 Transfer Function Sistem yang Direncanakan

Mengacu pada pembahasan di depan dapat diambil suatu bentuk transfer function dari sistem yang direncanakan. Dari diagram transfer function, sistem akan berubah akibat adanya pengurangan oleh W_{fu} , dan perubahan tersebut dikompensasi dengan memberi feedback W_{mf} terhadap nilai W_{uf} dan W_{ta} .



Gambar 5.2
Transfer Function Sistem yang Direncanakan

Input transfer function tersebut berupa posisi titik berat yang diinginkan, sedangkan outputnya berupa posisi titik berat setelah adanya kompensasi akibat perubahan berat bahan bakar atau perubahan titik berat pesawat terbang.

Jadi dari transfer function tersebut output akan menyesuaikan dengan input terhadap pemakaian bahan bakar (W_{fu}) dengan cara merubah besar W_{uf} dan W_{ta} sebesar W_{mf} , dimana W_{mf} adalah berat air yang harus dipindahkan untuk kompensasi. Dengan kata lain sistem tersebut bekerja dengan parameter komponen berat yang direncanakan terhadap perubahan variabel berat bahan bakar.

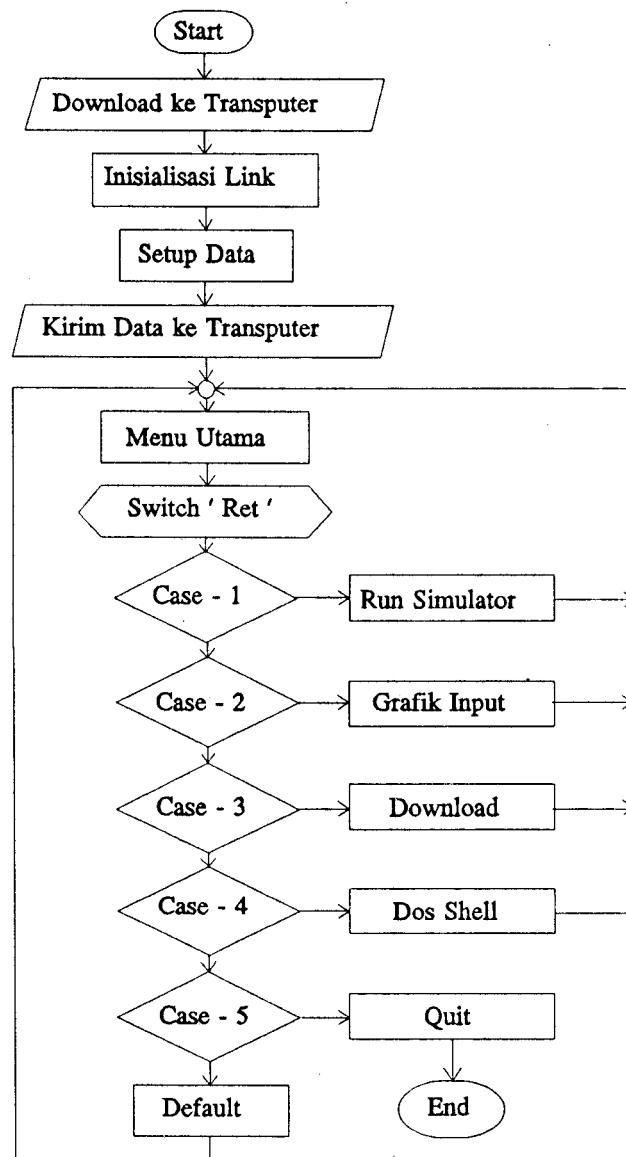
5.2 Pogram Utama di Komputer (PC)

Program yang dioperasikan di PC merupakan program induk dalam simulator ini dan berfungsi untuk menginputkan data, mengontrol sekaligus menampilkan hasil. Semua nilai komponen berat W_{AC} , W_F , W_{TF} , W_{TA} dan jarak X_{AC} , X_F , X_{TF} , X_{TA} serta CG_{INT} diset terlebih dahulu. Dari semua parameter ini selanjutnya dikirim ke transputer.

Apabila diinginkan merubah nilai parameter tersebut di atas, dapat dilakukan dengan meng-Edit data (parameter) yang terdapat di menu utama. Data hasil perubahan dikirimkan ke transputer dengan Download. Adapun Flowchart program utama di PC seperti pada gambar 5.3.

Penjelasan flowchart program utama :

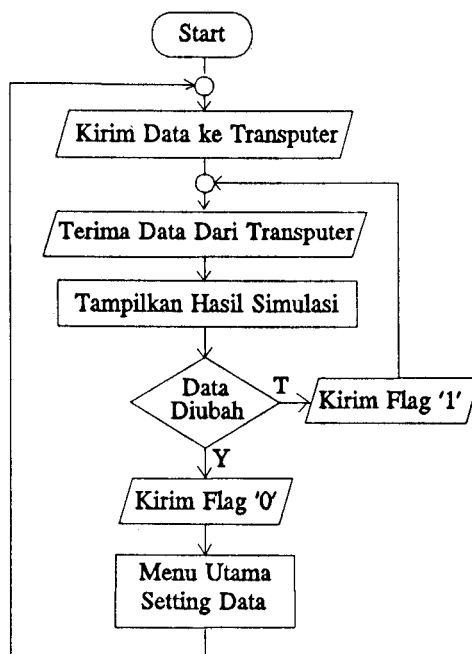
- ✓ Pada awal program dilakukan download program dengan *extension* TLD ke transputer, program yang didownload ini akan beroperasi di transputer untuk mengambil dan mengolah data.
- ✓ Kemudian dilakukan inisialisasi Link yang digunakan sebagai media komunikasi antara PC dan transputer.
- ✓ Setting data digunakan untuk menentukan harga awal dari semua parameter yang digunakan, dalam perencanaan ini parameter berat diset maximum.
- ✓ Data yang sudah diset dikirim ke transputer melalui Link, dan nantinya digunakan untuk mengolah data.
- ✓ Menu utama diperlukan sebagai inteface antara simulator dengan operator. Menu utama memberi return value sebagai kode pilihan menu ke program



Gambar 5.3
Flowchart Program Utama

utama. Return value dari menu utama akan di-check untuk menentukan operasi yang akan dilakukan, hal ini perlu untuk menentukan operasi yang sesuai dengan kehendak operator.

5.2.1 Menampilkan Simulasi



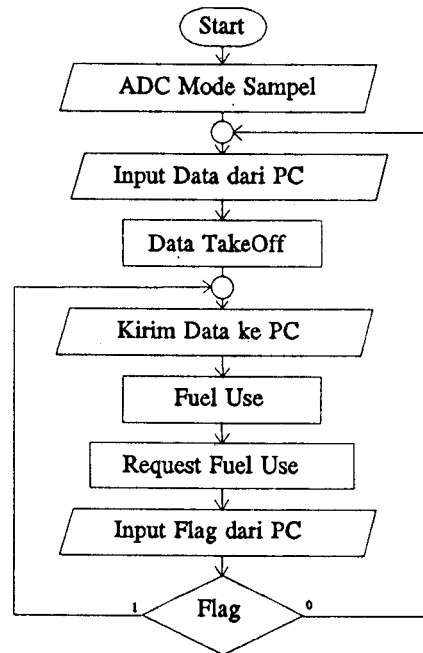
Gambar 5.4
Flowchart Simulasi di PC

Simulasi ditampilkan oleh PC pada monitor dengan mode VGA high, dan data hasil simulasi disimpan dalam disk. Selama menampilkan terjadi passing data hasil simulasi dari transputer ke PC dan sebuah karakter sebagai flag dari PC ke transputer, masalah komunikasi akan dibahas pada bab 5.4.

Data yang ditampilkan meliputi jumlah masing-masing komponen berat, posisi titik berat yang dikehendaki dan posisi titik berat akibat pemakaian bahan bakar serta titik berat hasil kompensasi.

Flowchart program simulasi di PC seperti pada gambar 5.4, terlihat program diatur sedemikian rupa sehingga akan terjadi kesesuaian passing data dengan transputer flowchart pada gambar 5.5, guna menghindari deadlock.

5.3 Program di Transputer



Gambar 5.5
Flowchart Simulasi di Transputer

Penjelasan flowchart program di transputer :

- ✓ Awal program di transputer mengkondisi modul ADC menjadi mode sampel, mengisi semua parameter yang diperlukan berharga nol.
- ✓ Menerima data dari PC sebanyak 10 data bertipe float atau 40 byte, data ini sebagai parameter dalam mengatur posisi titik berat terhadap perubahan berat bahan bakar.
- ✓ Data dari PC dihitung terlebih dahulu guna mendapatkan data saat take off.
- ✓ Mengambil data dari modul ADC sebagai simulasi besar aliran pemakaian bahan bakar, data tersebut dikonversikan kedalam satuan Kg/s. Kemudian menghitung berat pemakaiannya dengan cara mengalikan besar aliran terhadap waktu

pemakaian.

- ✓ Menghitung berat air yang harus dipindahkan dari salah satu tangki ke tangki lainnya untuk mengantisipasi perubahan titik berat, berdasarkan berat bahan bakar yang dipakai.
- ✓ Data hasil perhitungan dikumpulkam dalam suatu buffer, selanjutnya dikirimkan ke PC untuk ditampilkan.
- ✓ Menerima sebuah flag dari PC apakah ada perubahan parameter, jika ada transputer akan menerima data baru dari PC tetapi jika tidak proses perhitungan dengan parameter lama dilanjutkan.

5.3.1 Mengambil Data ADC

Modul ADC dihubungkan dengan transputer pada Link 1, mempunyai empat sinyal kontrol yaitu stconv (start conversi), endconv (end conversi), hold dan reset. Masing-masing sinyal kontrol mempunyai fungsi sebagai berikut:

- stconv : perintah supaya ADC memulai mengkonversi sinyal input.
- endconv : perintah supaya sinyal kontrol start konversi ADC tidak aktif.
- hold : berfungsi mengaktifkan rangkaian sample hold.
- reset : mengembalikan ADC dan sample hold ke kondisi tidak aktif.

Langkah-langkah pengambilan data:

- Mengaktifkan sample hold sehingga level sinyal input dipertahankan.
- Mengirim stconv = '0', sewaktu terjadi perubahan sinyal dari '1' menjadi '0' pada pin -WR, ADC mulai mengkonversi sinyal input, dilanjutkan dengan

mengambil range.

- Mengirim endconv = '1', sehingga logika di pin -WR ADC berlogika '1' atau siap mengkonversi.
- Setelah itu menunggu sampai pin INTR ADC berlogika '1' ditandai dengan status data input valid (IValid berlogika '1').
- Membaca hasil konversi ADC sebanyak 8 bit.
- Mengembalikan status sample hold dan semua kontrol ADC ke keadaan awal.
- Menggabungkan antara range dan hasil konversi menjadi data hasil konversi sebenarnya.

Prosedur pengambilan data ADC seperti di bawah ini.

```
void GetDataFromLink (void)
{
    unsigned int data;
    unsigned char range, step;
    char stconv = 0,
        endconv = 1,
        hold = 1,
        reset = 3;

    while (TRUE)
    {
        ChanOutTimeFail (LINK0OUT + link, &hold, 1, 160+Time());
        ChanOutTimeFail (LINK0OUT + link, &stconv, 1, 160+Time());
        ChanInTimeFail (LINK0IN + link, (char *)&range, 1, 160+Time());
        ChanOutTimeFail (LINK0OUT + link, &endconv, 1, 160+Time());
        ChanInTimeFail (LINK0IN + link, (char *)&step, 1, 160+Time());
        ChanOutTimeFail (LINK0OUT + link, &reset, 1, 160+Time());
        data = 2*((range&7)*256 + step);
    }
}
```

5.3.2 Mengolah Data

Data ADC digunakan sebagai data aliran bahan bakar yang dibutuhkan oleh pesawat terbang waktu beroperasi dalam satuan Kilogram per secon (Kg/s). Maksimum aliran bahan bakar 20,6 Kg/s sama dengan data 4094 dari ADC. Dengan mengetahui laju aliran bahan bakar, dapat diketahui jumlah pemakaian

bahan bakar selama waktu tertentu,

$$\text{jumlah pemakaian} = \text{besar aliran} * \text{waktu.}$$

Awal proses setelah menerima data dari PC adalah menghitung dan membagi air pada tangki depan dan tangki belakang agar posisi titik berat sesuai yang diinginkan. Dilanjutkan dengan menghitung berat total, momen total untuk mendapatkan jarak titik berat saat take-off menggunakan persamaan (2.4).

Data dari ADC dipakai untuk mencari berat bahan bakar yang digunakan (W_{FU} : Weight Fuel Use) selama beroperasi, jadi berat totalnya berkurang sebesar W_{FU} . Titik berat yang baru dicari dengan menggunakan persamaan (2.7), dan untuk mengkompensasi digunakan persamaan (2.13). Berubahnya berat akan mempengaruhi titik beratnya sehingga diperlukan penyelesaian persamaan (2.20) untuk mengantisipasi perubahan titik berat.

5.4 Komunikasi antara PC dan Transputer

Hal penting komunikasi antara PC dan transputer meliputi tipe data dan jumlah data, sebab apabila kedua hal tersebut tidak sesuai akan terjadi *deadlock*. *Deadlock* bisa juga terjadi apabila antara PC dan transputer sama-sama siap mengirim atau sama-sama siap menerima. Cara menghindari *deadlock* diantaranya harus dipastikan bahwa selama PC mau mengirim maka transputer harus siap menerima atau sebaliknya, juga perlu diperhatikan jumlah data yang ditransfer.

Perlu diketahui panjang dari tiap tipe data antara transputer dan PC karena ada sedikit perbedaan. Misalkan tipe **Char** baik di transputer maupun di PC panjangnya 1 byte (8 bit), tetapi tipe **Int** kalau di PC panjangnya 2 byte (16 bit)

lain lagi kalau di transputer karena panjangnya 4 byte (32 bit).

Supaya transfer data lebih baik perlu dilakukan pengetesan status link, jadi sebelum mengirim atau menerima link ditest lebih dulu apakah statusnya sudah valid, jika sudah pengiriman atau penerimaan data dapat dilakukan. Tetapi jika belum valid maka dapat menunggu sampai valid atau dapat ditinggal untuk mengerjakan proses lainnya.

BAB VI

PENGUKURAN DAN PENGUJIAN

Sesudah dilakukan perencanaan hardware dan software maka sebelum dijalankan sebagai suatu sistem tiap-tiap bagian perlu dilakukan pengujian terlebih dahulu. Bila rangkaian telah bekerja, tahap selanjutnya adalah pengukuran dan sekaligus pengujian.

6.1 Pengukuran Alat

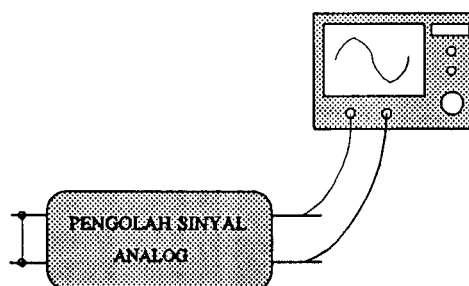
Untuk merealisasikan sistem yang direncanakan maka selanjutnya dilakukan pengukuran dan pengujian alat. Pengujian dilakukan dengan cara mengukur dan mencoba peralatan yang dibuat. Pengukuran dilakukan pada bagian-bagian yang memerlukan penyetelan.

Pengukuran dan pengujian dilakukan di ruang B203 R&D Laboratory Bidang studi Elektronika, dengan menggunakan oscilloscope Hawlet Packard 1740A, DC power supply Hawlet Packard 6200B, function generator Hawlet Packard 3311A & digital multimeter Hitech HT-8000.

6.1.1 Pengukuran Noise

Pengukuran noise dilakukan dengan menghubungkan-singkatkan kedua input. Selanjutnya diukur tegangan noise pada outputnya dengan menggunakan osiloskop. Dari pengukuran ini diperoleh tegangan noise sebesar $2,5 \text{ mV}_{pp}$. Karena range tegangan output pengolah sinyal analog mencapai 4094 mV_{pp} , maka S/N rasionya

adalah 1638 ($4094/2,5 = 1638$).



Gambar 6.1
Pengukuran Noise

6.1.2 Pengukuran Auto Range

Tegangan referensi dari zener sebesar 5 V diatur sedemikian sehingga tegangan referensi komparator pertama atau tertinggi sebesar 3584 mV dengan cara mengatur besar resistor variabel V_8 . Untuk komparator berikutnya digunakan resistor divider masing-masing 1 KOhm untuk masing-masing range sehingga mempunyai kelipatan 512 mV ($3584 \text{ mV}/7 = 512 \text{ mV}$).

Untuk menguji kebenaran kerja auto range ini, VI_COMP diberi tegangan DC 0 mV kemudian tegangan output komparator diukur dengan menggunakan multimeter digital, didapat $VO_COMP = 0 \text{ mV}$. Tegangan input dinaikkan, VO_COMP masih berharga 0 mV, hasil VO_COMP baru berubah dari 0 mV menjadi 512 mV saat tegangan input berubah melebihi 512 mV. Tabel hasil perbandingan tegangan input (VI_COMP) dengan tegangan output (VO_COMP) sebagai berikut:

Tabel 6.1
Hasil Pengukuran Auto Range

No.	V Input (mV)	Vo Comp (mV)	V out (mV)
1.	0	0	0
2.	200	0	200
3.	400	0	400
4.	600	512	88
5.	800	512	288
6.	1000	512	488
7.	1200	1024	176
8.	1400	1024	376
9.	1600	1536	64
10.	1800	1536	264
11.	2000	1536	464
12.	2200	2048	152
13.	2400	2048	352
14.	2600	2560	40
15.	2800	2560	240
16.	3000	2560	440
17.	3200	3072	128
18.	3400	3072	328
19.	3600	3584	16
20.	3800	3584	216
21.	4000	3584	416

6.2 Pengukuran Modul ADC 0804

ADC 0804 bila mendapat tegangan setengah referensi 256 mV akan memiliki rentang 512 mV pada keadaan skala penuh, dimana hal ini berarti ketelitian 1 bit sama dengan 2 mV. Dengan metoda range yang telah dijelaskan di depan, dengan ketelitian tetap dipertahankan pada 2 mV per 1 bit data, maka modul hardware ini dapat mencapai rentang konversi tegangan input 0 mV sampai 4094 mV.

Start konversi dilakukan secara software. Hal ini berkaitan dengan sinkronisasi data yang diambil dari sistem ADC. Komunikasi antara modul ADC dengan transputer melalui link adaptor (mode 1), dimana *handshaking* hanya ada

empat bit yaitu dua bit untuk input (*IValid* dan *IAck*) dan dua bit untuk output (*QValid* dan *QAck*). Start konversi dilakukan dengan cara mengirimkan sinyal '0' ke pin -WR, dan kemudian dijadikan '1'. Antara sinyal '0' dan sinyal '1' tadi dimanfaatkan untuk mengambil data range saat itu, agar didapatkan harga range yang tepat. Bila pengambilan harga range terlambat menyebabkan hasil konversi tidak sesuai. Setelah waktu konversi selesai data dapat diambil untuk diproses lebih lanjut. Jadi pengaturan start konversi secara manual akan menghindari terbaliknya susunan data, antara data range dengan data hasil konversi. Berikut ini adalah softwre untuk test modul ADC.

```
#include <stdio.h>
#include <stdlib.h>
#include <conc.h>
#define TRUE 1
#define FALSE 0
char reset = 3;

void GetDataFromLink(int link)
{
    unsigned char range, step;
    unsigned int data;
    char stconv = 0;
    char endconv = 1;
    char hold = 1;
    while (!kbhit())
    {
        ChanOutTimeFail (LINK0OUT + link, &hold, 1, 160+Time());
        ChanOutTimeFail (LINK0OUT + link, &stconv, 1, 160+Time());
        ChanInTimeFail (LINK0IN + link, (char *)&range, 1, 160+Time());
        ChanOutTimeFail (LINK0OUT + link, &endconv, 1, 160+Time());
        ChanInTimeFail (LINK0IN + link, (char *)&step, 1, 160+Time());
        ChanOutTimeFail (LINK0OUT + link, &reset, 1, 160+Time());
        data = ((range&7)*512 + step*2);
        printf (" %2d : %3d : %4d \r", (range&7), step, data);
    }
}

void main(void)
{
    int link_id, status;
    while(TRUE)
    {
        status=FALSE;
        while(!status)
        {
            printf("Channel pada link ke (1-3): ");
            scanf("%d",&link_id);
            if (link_id>=1 && link_id<=3) status=TRUE;
        }
        printf("\n");
        printf("Data dibaca dari link : %d\n",link_id);
        ChanOutTimeFail(LINK0OUT+link_id,&reset,1,160+Time());
        GetDataFromLink(link_id);
    }
}
```


Program tersebut di atas setelah di compile menjadi file berekstensi .TLD, di download ke transputer dan di PC dijalankan program CIO.EXE. Dengan menggunakan oscilloscope dilihat pada pin S.HOLD, WR, INTR, DTA1, DTA2. Terlihat logika dari masing-masing pin tersebut sesuai dengan rancangan yang dikehendaki.

6.3 Serial Link IMS C011 Mode 1

IMS C011 beroperasi pada mode 1 dengan tidak menghubungkan pin *SeparateIQ* ke ground. Ada empat bit data untuk mengatur sinkronisasi lalu lintas data yang akan masuk maupun keluar dari transputer.

Saat input, bila data valid, pin Ivalid diset '1', disini menggunakan IC 74LS74 (D flip-flop). Setelah data tersebut diambil oleh transputer, dikirim sinyal pemberitahuan melalui pin *IAck*. Sinyal ini digunakan untuk mereset 74LS74 (pin *IInvalid*) agar outputnya menjadi '0', sehingga link adaptor siap untuk menerima data lagi.

Pengaturan saat output lebih mudah, yaitu dengan menghubungkan pin *QValid* dengan pin *QAck*. Dengan begitu setiap mengirimkan data, *QValid* diset '1' yang berakibat *QAck* mendapat sinyal '1' juga, sehingga link siap mengirimkan data kembali. Gambar lengkap rangkaian dapat dilihat pada lampiran.

Dengan menjalankan program test modul ADC di atas didapat hasil yang ditampilkan di layar sebagai berikut:

Tabel 6.2
Hasil Konversi Modul ADC

No.	V Input (mV)	Range ke-	Data ADC	Data (mV)
1.	0	0	0	0
2.	200	0	99	198
3.	400	0	200	400
4.	600	1	42	596
5.	800	1	144	800
6.	1000	1	244	1000
7.	1200	2	86	1194
8.	1400	2	188	1400
9.	1600	3	32	1600
10.	1800	3	133	1802
11.	2000	3	232	2000
12.	2200	4	75	2198
13.	2400	4	177	2402
14.	2600	5	18	2596
15.	2800	5	119	2798
16.	3000	5	221	3002
17.	3200	6	65	3202
18.	3400	6	164	3400
19.	3600	7	6	3596
20.	3800	7	107	3898
21.	4000	7	209	4002

6.4 Pengujian Sistem

Seperti telah dijelaskan di depan bahwa parameter yang digunakan dalam simulator ini terdiri komponen berat dan jaraknya dari datum, W_{AC} , W_F , W_{TF} , W_{TA} , L_{AC} , X_F , X_{TF} , X_{TA} serta titik berat yang diinginkan CG_{INIT} , dengan mengabaikan parameter lainnya.

Pengujian ini bertujuan untuk mengetahui batas minimum dan maksimum dari pengaturan CG & WMS. Pengujian ditinjau dari kemampuan pengaturan beban pada saat take-off, apabila pada saat take-off pengaturan beban bisa menempatkan posisi titik berat atau CG_{TO} sesuai dengan CG_{INIT} maka pada kondisi

ini CG & WMS bekerja dengan baik. Tetapi apabila CG_{TO} tidak sesuai dengan CG_{INIT} maka CG & WMS tidak mampu mengadakan pengaturan dengan baik.

Dengan memberi nilai parameter-parameternya sebagai berikut didapat:

$$W_{AC} = 15700 \quad L_{AC} = 26,568$$

$$W_F = 4200 \quad X_F = 7,025$$

$$W_{TF} = 780 \quad X_{TF} = 5,718$$

$$W_{TA} = 0 \quad X_{TA} = 18,718$$

Tabel 6.3
Kondisi Take-off pada Berbagai Harga CG_{INIT}

No.	CG_{INIT} (%)	W_{TF} (Kg)	W_{TA} (Kg)	CG_{TO} (%)
1.	24	7800	0	23,64
2.	25	7800	0	24,4
3.	26	746,52	33,47	26
4.	27	644,74	135,25	27
5.	28	542,97	237,02	28
6.	29	441,19	338,80	29
7.	30	339,41	440,58	30
8.	31	237,64	542,35	31
9.	32	135,86	644,13	32
10.	33	34,09	745,90	33
11.	34	0	7800	33,83
12.	35	0	7800	34,59

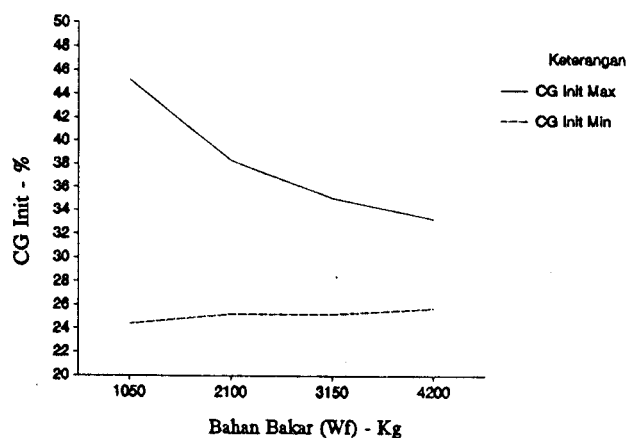
Dari tabel hasil pengujian di atas terlihat bahwa titik berat pesawat terbang pada waktu take-off (CG_{TO}) akan sama dengan titik berat yang diinginkan (CG_{INIT}) apabila besar CG_{INIT} 26% -33% lebih tepatnya 25,7% - 33,3%. Jadi apabila CG_{INIT} diset diantara harga tersebut maka CG & WMS akan dapat mengatur titik berat dengan baik.

Pada berbagai jumlah bahan bakar (W_F) pada saat take-off, dengan pengujian yang sama untuk mendapatkan pengaturan yang baik didapat batas maximum dan minimum CG_{INIT} sebagai berikut.

Tabel 6.4
Batas CG_{INIT} Pada Berbagai Jumlah W_F

No.	W_F (Kg)	CG_{MIN} (%)	CG_{MAX} (%)
1.	1050	24,4	45,2
2.	2100	25,2	38,3
3.	3150	25,5	35,1
4.	4200	25,7	33,3

Terlihat dari tabel di atas bahwa semakin besar jumlah bahan bakar, batas minimum semakin naik dan batas maximum semakin turun atau antara kedua batas semakin sempit. Jadi semakin besar jumlah bahan bakar pada saat take-off semakin berkurang kemampuan CG & WMS untuk mengatur posisi titik berat pesawat terbang. Hal ini disebabkan pembatasan kapasitas air yang dipakai sebagai beban dalam ballast system.



Gambar 6.2
Batas CG_{INIT} Pada Berbagai Jumlah Bahan Bakar

BAB VII

PENUTUP

7.1 Kesimpulan

Dari tugas akhir yang telah diselesaikan dapat diambil kesimpulan sebagai berikut:

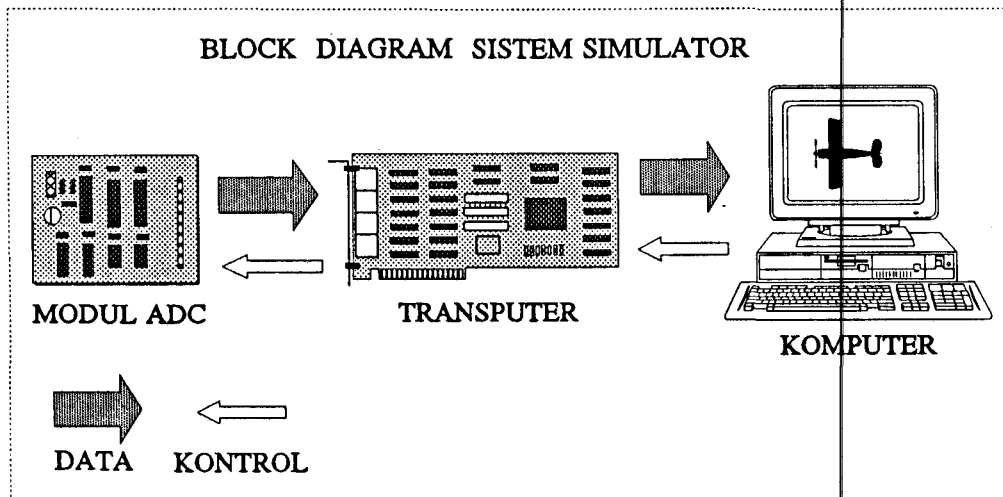
- Simulator CG & WMS berfungsi membantu atau mempermudah menganalisa proses suatu system pengaturan perubahan titik berat pesawat terbang akibat pemakaian bahan bakar.
- Semakin bertambah jumlah bahan bakar pada saat take-off batas kemampuan CG & WMS untuk mengatur posisi titik berat pesawat terbang semakin berkurang, disebabkan terbatasnya kapasitas air yang dipakai dalam ballast system.
- Pemakaian transputer dilengkapi dengan Floating Point Unit (FPU) membantu mempercepat perhitungan bilangan real (float), dan dengan distribusi program pada transputer dan PC (*Distributedprocessing*) membantu proses simulasi.

7.2 Saran

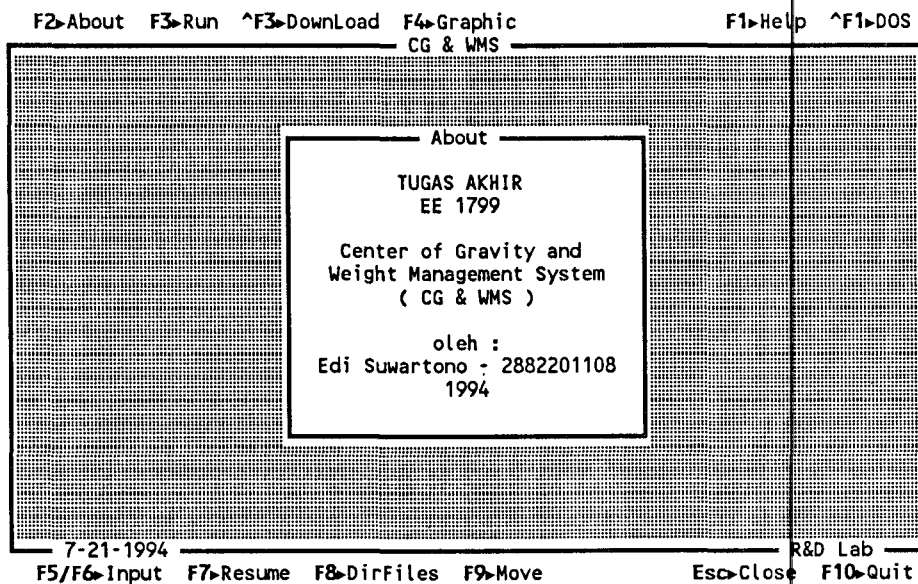
Proses simulasi dapat dipercepat dan ditingkatkan unjuk kerjanya dengan memanfaatkan konfigurasi paralel transputer dan atau memakai metode yang lebih baik misalkan Neural Network (jaringan syaraf tiruan).

DAFTAR PUSTAKA

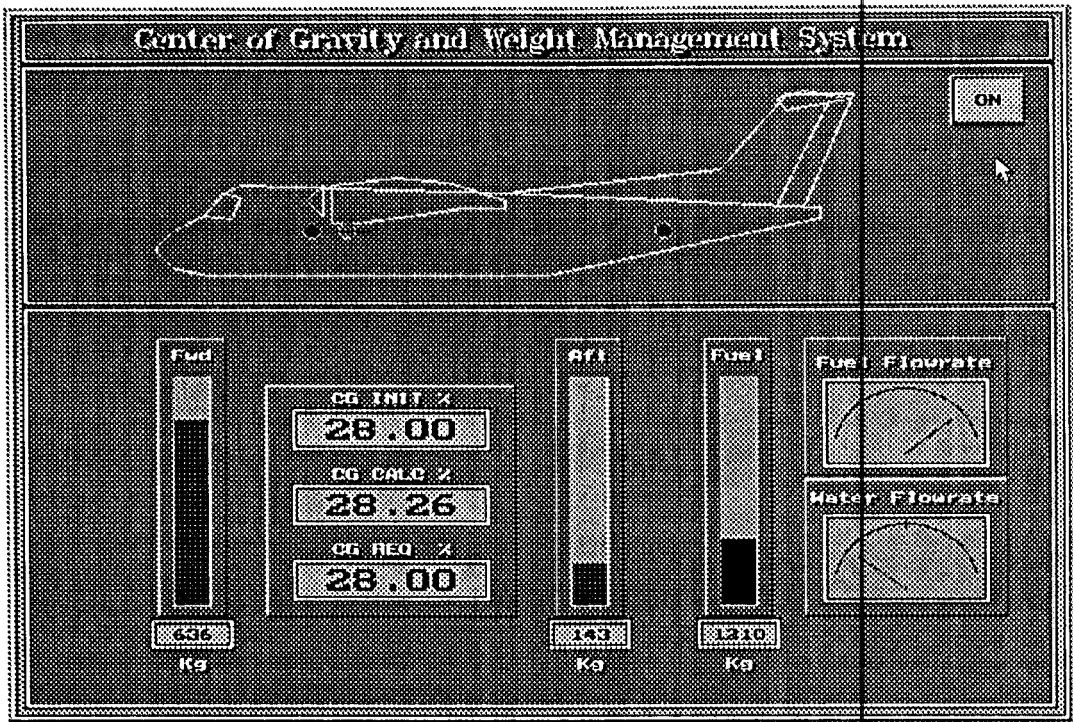
1. Gleaser, Dennis, *An Invitation to Fly: Basic for the Private Pilot*, Second Edition, USA.
2. Graham, Ian, *The Transputer Handbook*, Printice Hall Intenational (UK) Ltd, 1990.
3. Kadir, Abdul, *Pemrograman Dasar Turbo C untuk IBM PC*, Edisi pertama, cetakan pertama, Andi Offset Yogyakarta, 1991.
4. Nelson, Robert C., *Flight Stability and Automatic Control*, Mc Graw Hill, Inc. 1990.
5. Sanderson, Jeppensen, *Aviation Fundamental*, 1988.
6. Shevell, Richard S., *Fundamental of Flight*, Secon Edition, Printice Hall International Inc., USA, 1989.
7. Sjartuni, Ananta, *Lebih Lanjut dengan C*, Edisi pertama, cetakan pertama, Elex Media Komputindo Jakarta, 1992.
8. _____, *Engineering Data IMS C011 Serial Link Interface*, INMOS Limited, USA, October 1988.
9. _____, *Linear Databook 2*, National Semiconductor Corporation Rev.1, 1988 Edition, 1988.
10. _____, *Transputer Education Kit, Transputer Architecture and Overview*, Computer System Architects (CSA), USA, 1990.
11. _____, *Transputer Education Kit, Logical System C89.1*, Computer System Architects (CSA), USA, 1990.
12. _____, *Transputer Education Kit, Occam and the Transputer A Workbook*, Computer System Architects (CSA), USA, 1990.
13. _____, *Transputer Education Kit, User Guide, Theory of Operation, Installation, Schematics*, Computer System Architects (CSA), USA, 1990.
14. _____, *Transputer Technical Notes*, INMOS Limited, Printice Hall Intenational (UK) Ltd, 1989.



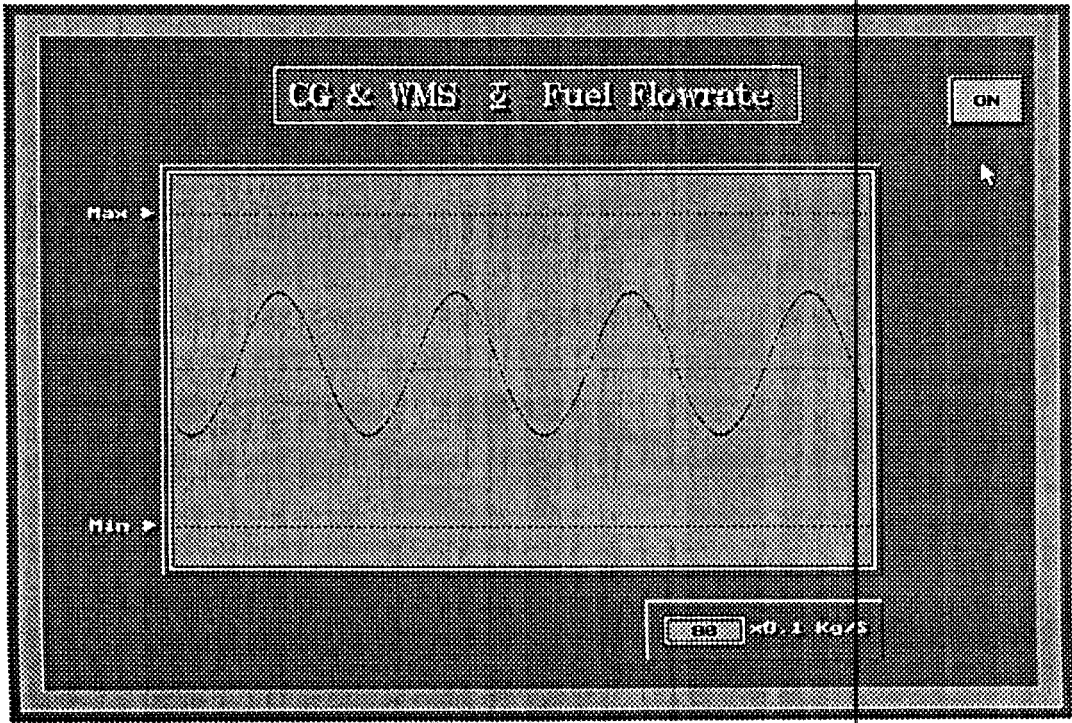
Block Diagram System



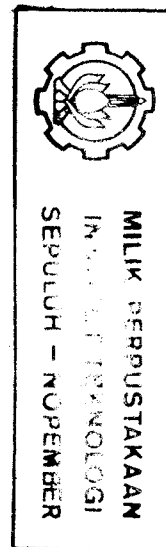
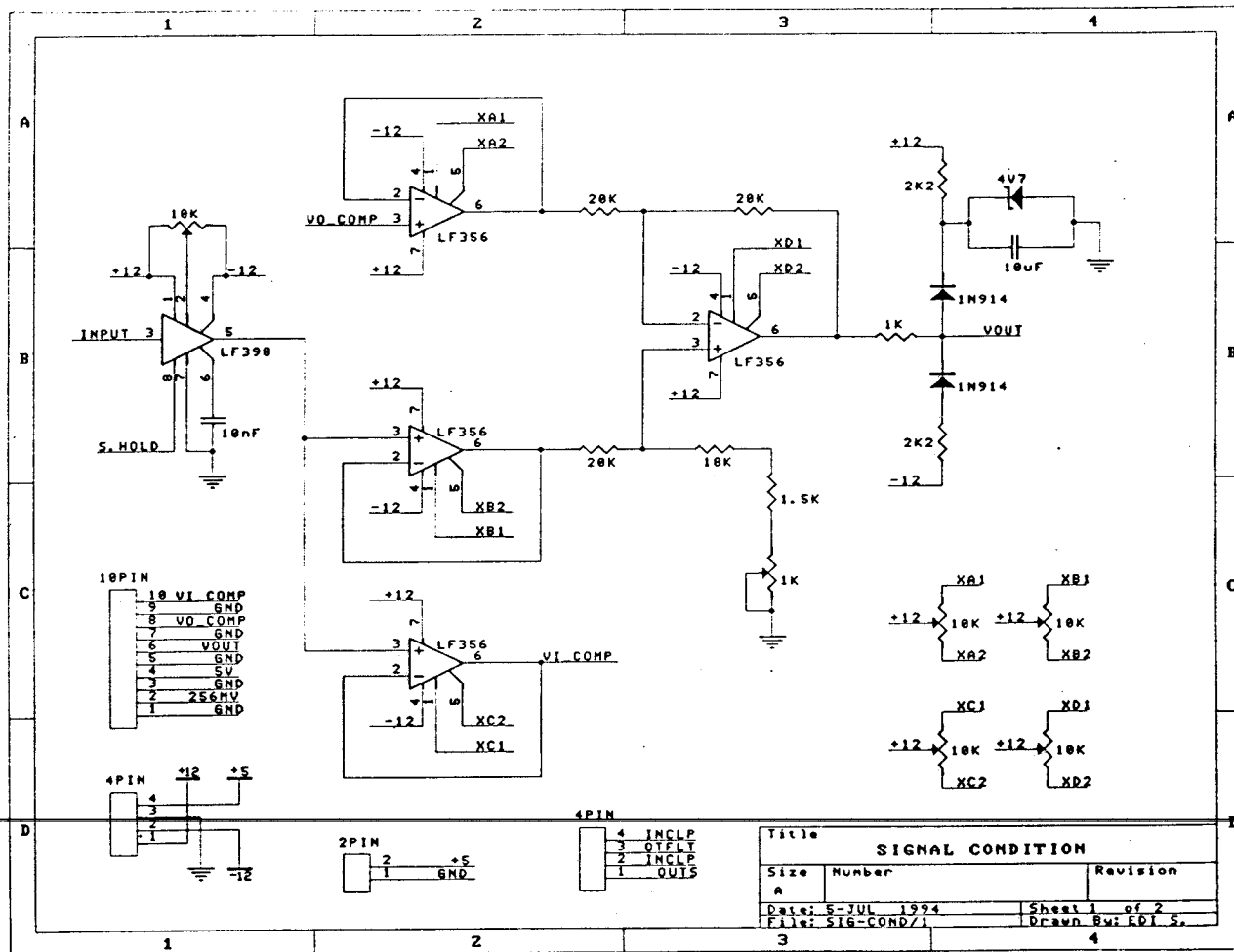
Tampilan Menu Utama

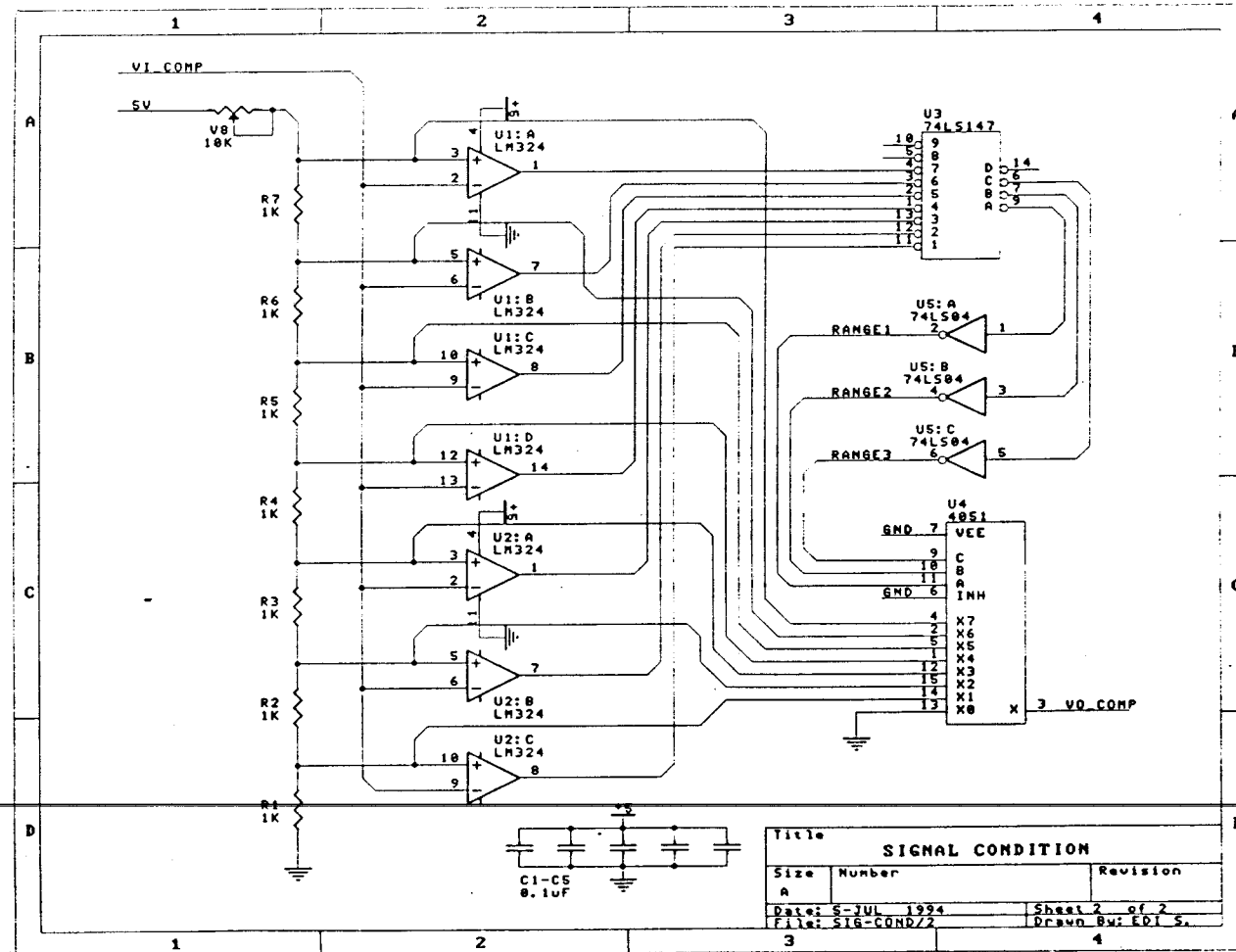


Tampilan Selama Proses Simulasi

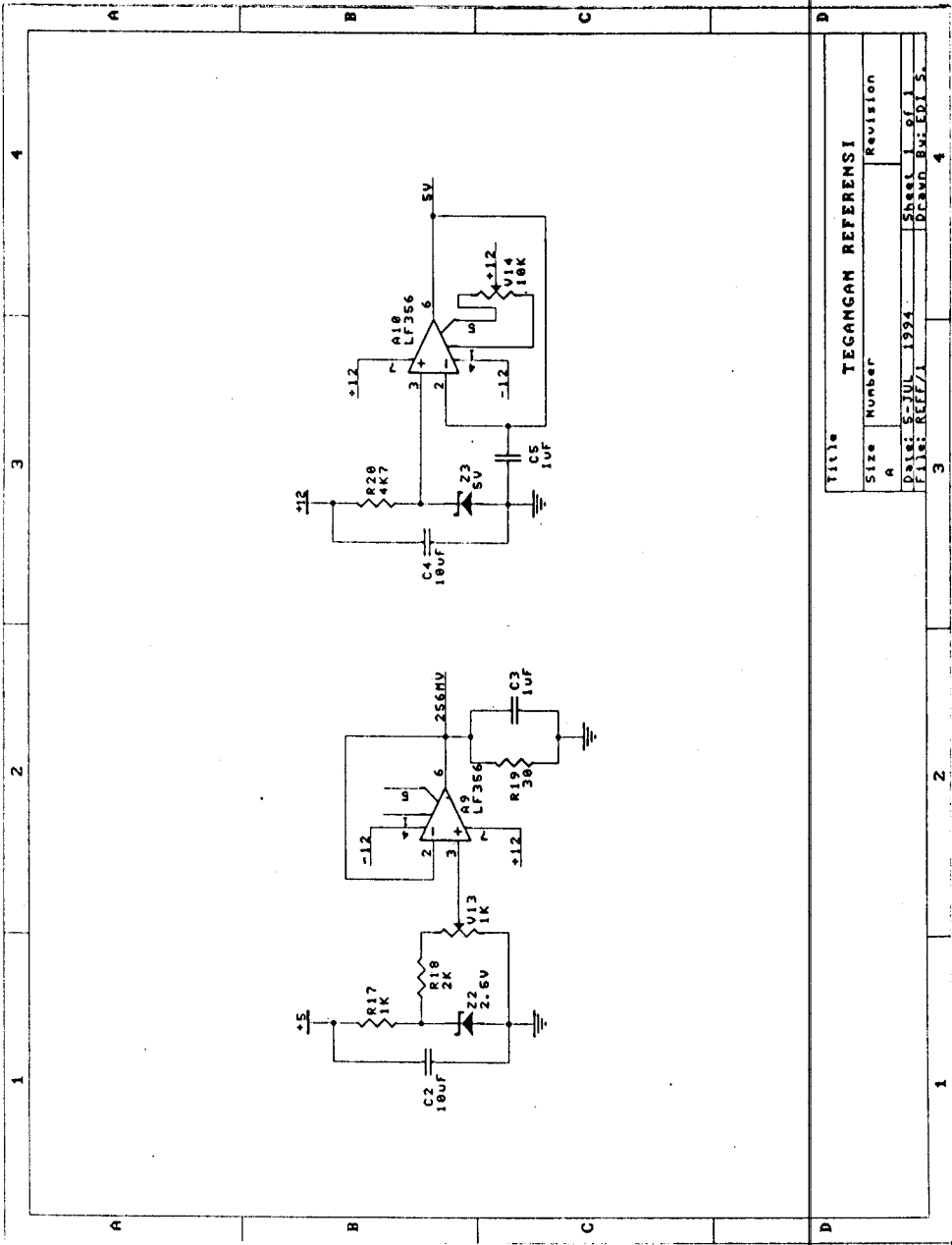


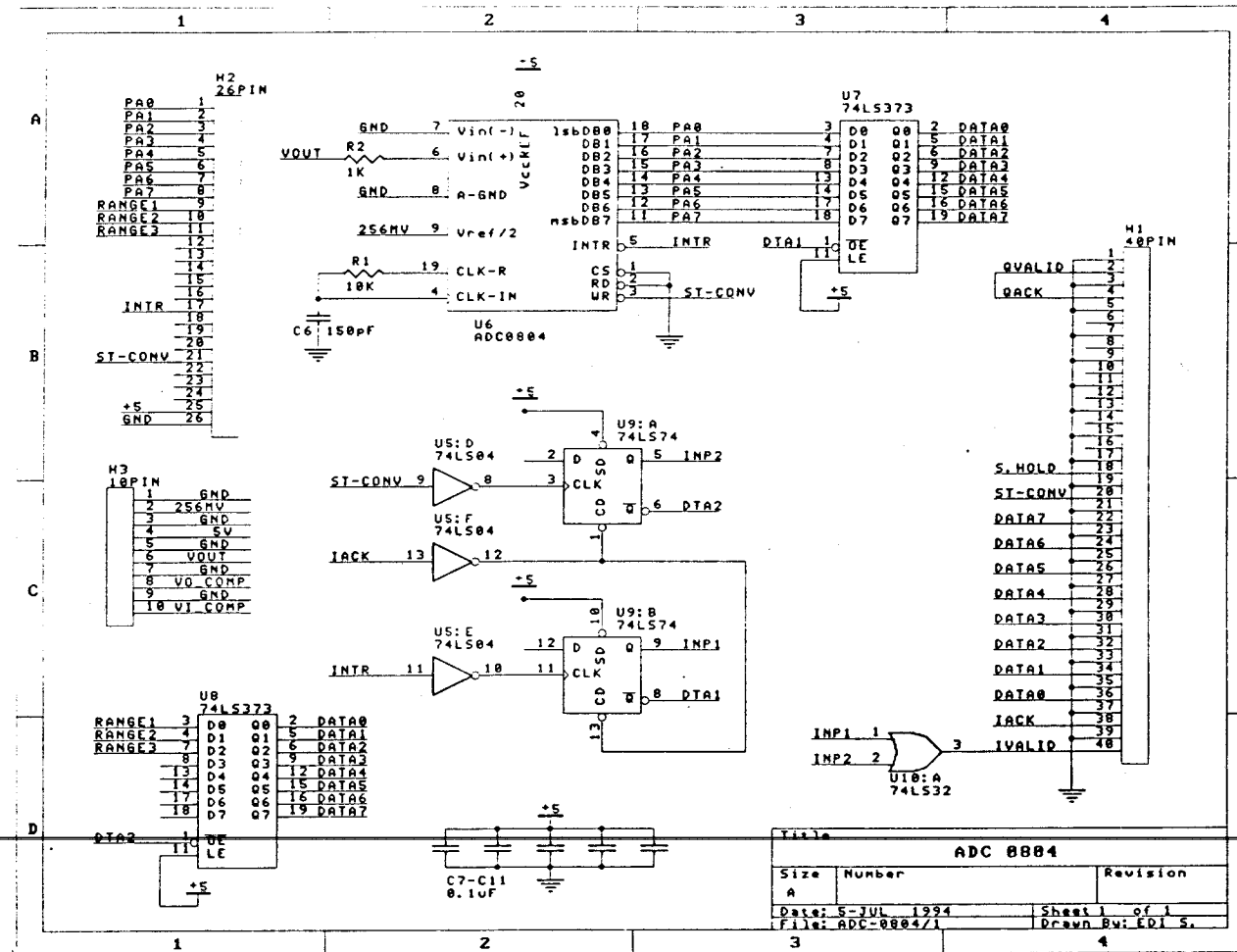
Tampilan Hasil Pembacaan ADC





Lampiran B-2





Title		
ADC 8884		
Size	Number	Revision
A		
Date: 5-JUL-1994		Sheet 1 of 1
File: ADC-8884/1		Drawn By: EDI S.

Lampiran B-4

